(2)

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A PROGRAM FOR SCHEDULING A
PATROL AIR WING TRAINING PLAN

by

David V. Hutson

June 1988

Thesis Advisor: Neil C. Rowe

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) Code 52-CS
7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)
8c ADDRESS (City, State, and ZIP Code)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification)
A PROGRAM FOR SCHEDULING A PATROL AIR WING TRAINING PLAN

12 PERSONAL AUTHOR(S)
Hutson, David V.

13a TYPE OF REPORT Master's Thesis	13b TIME COVERED FROM _____ TO _____	14 DATE OF REPORT (Year, Month, Day) 1988 June	15 PAGE COUNT 188

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Schedule, Artificial Intelligence, Prolog, Training, Air Wing, Decision Support, Knowledge Base, Search,

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This research examined the feasibility of a computerized scheduling system to assist the development of an annual training plan for a Patrol Air Wing. A prototype is proposed incorporating a modified A* search control structure to handle the combinatorial part of the problem. The system uses a pre-existing file for its database and is implemented on an ISI workstation using the Prolog computer language. Comparisons with a manual derivation of the training plan are made and analysis of the prototype results with the pruning variable at several levels is performed.

A-Star!

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. Neil C. Rowe	22b TELEPHONE (Include Area Code) 408-646-2462 / 22c OFFICE SYMBOL Code 52Rp

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted.
All other editions are obsolete

Approved for public release; distribution is unlimited.

A PROGRAM FOR SCHEDULING A PATROL AIR WING TRAINING PLAN

by

David V. Hutson
Lieutenant Commander, United States Navy
B.A., Harding University, 1972

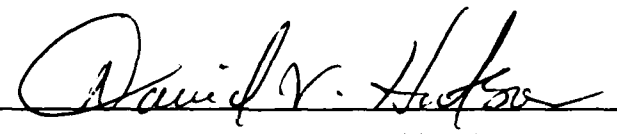
Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL

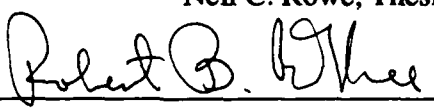
June 1988

Author: 

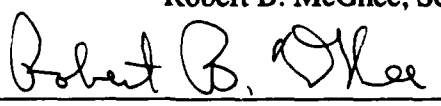
David V. Hutson

Approved by:

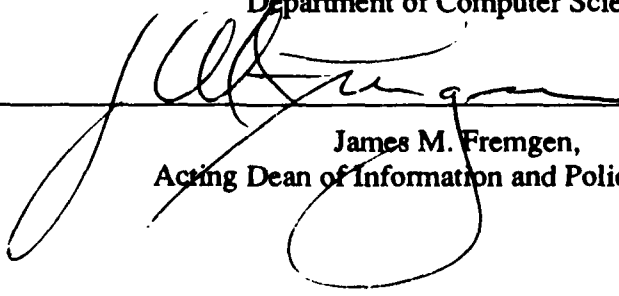
Neil C. Rowe, Thesis Advisor



Robert B. McGhee, Second Reader



Robert B. McGhee, Acting Chairman,
Department of Computer Science



James M. Fremgen,
Acting Dean of Information and Policy Sciences

ABSTRACT

This research examined the feasibility of a computerized scheduling system to assist the development of an annual training plan for a Patrol Air Wing. A prototype is proposed incorporating a modified A* search control structure to handle the combinatorial part of the problem. The system uses a pre-existing file for its database and is implemented on an ISI workstation using the Prolog computer language. Comparisons with a manual derivation of the training plan are made and analysis of the prototype results with the pruning variable at several levels is performed.

TABLE OF CONTENTS

LIST OF TABLES

ACKNOWLEDGEMENTS

Thesis preparation takes a lot of time, hard work, and dedication to turn out the best possible product. It also requires the cooperation of several people all to whom I extend my thanks:

LCDR George Sanford, past PATWING TEN Training Officer, who expertly detailed the scheduling procedures that my thesis strives to emulate.

Dr. Hank Smith, PATRON THIRTY-ONE Education Coordinator, who initially introduced me to the research topic and coordinated the necessary interviews with LCDR Sanford.

Chet Hartley, my friend, who dedicated countless hours toward editing and formatting my thesis text.

Kelley, my beautiful wife, who understood the importance of my work and consistently demonstrated her love through unselfish support.

I. INTRODUCTION

In a military environment the turnover of command personnel often hurts the corporate knowledge within that command. Turnover letters, regulations and instructions attempt to provide means for a smooth transition. These documents contain quick reference for guidance, initial briefing information, and descriptions of problems that deserve special attention. However, certain billets require more than these historical precautions. These billets require a vast amount of knowledge regarding the job-related tasks that cannot readily be extracted from the file of turnover information for immediate practical application. Thus the command experiences a performance lag as the new job holder gradually builds the knowledge base necessary to make decisions correctly and with confidence. This performance lag is not usually related to the individual's effort.

There is a method for minimizing the trial-and-error period. This method can also prevent the usual performance lag associated with sophisticated decision making billets and is especially useful if the tasks involve management of a matrix of multiple resources and recipients. It is computerized decision support for scheduling. The long period associated with developing the necessary knowledge base can be significantly abbreviated by a computer system that incorporates most of that knowledge base.

One such billet that can serve as a model for this system is that of the training officer of a intermediate level command. In this study this will be Commander, Patrol Wing Ten at Naval Air Station Moffett Field, California (COMPATWING TEN). The Training Officer is charged with scheduling the required inspections for each of the seven squadrons under COMPATWING TEN's cognizance.

Scheduling seven squadrons involves management of internal and external resources. O'Brien [Ref. 1] believes the most important scheduling resource is time. The training officer manages numerous consumers of time. With time as a limited resource, the project becomes a challenging decision making problem: the solution must cope with subproject time constraints as well as resource constraints. This research examines the schedule for fiscal year 1986 and at one early stage there are eighty-nine possible paths. With a forty member solution, this is such a big combinatorial problem that almost all present methods for finding optimal schedules are impractical for a computer [Ref. 2]. The problem is also frustrating since it is fairly easy to state and visualize: once resolved, a bar graph easily communicates the solution. The frustration lies in finding an optimization scheme that gets an answer in a reasonable amount of time.

A training plan is scheduled over one fiscal year. Each squadron has an eleven month window in which its training must occur. In most cases, this occurs in more than one fiscal year, due to the rotational nature of the deployed squadrons. Of the seven squadrons there are always at least two deployed. The remaining five are either just returning from a deployment, preparing to depart to a deployment site, or in the period of training availability. No training inspections can be conducted if a squadron is in a safety stand-down period or is just prior to deployment. Further constraints on the time available are scheduled assignments to the ready alert, a training event which consumes thirty to forty-five days for each assignment. Up to two such assignments can be scheduled per squadron. In addition, there are six different inspections scheduled per squadron. These inspections are of one to ten working days in duration, and must not be interrupted by holidays or other inspections. Certain ones must precede others and some require scheduling within an allotted period.

I built a prototype system for scheduling a training plan. Once all the desired enhancements are made, it can serve as a competent decision support tool. The Training Officer will only be required to input dates related to deployments and supply the file holding the previous year's schedule. The output will be a proposal that can either be accepted or modified. The major advantages are the time saved and the accuracy of the solution. This system is intended to supplement the turnover file and improve performance throughout the Training Officer's tour. During the process of building the prototype my objectives were:

- Show that the knowledge base necessary for decision making by an experienced scheduler can be gathered and translated for use in a prototype scheduling system.
- Demonstrate that Prolog is a suitable language for implementing a prototype scheduling system.
- Demonstrate that the prototype can attain different degrees of optimization in determining the solution.

This research proposes a search technique with a cost analysis and an agenda of possible next states in working toward the solution. The time to arrive at a solution, the computer storage space required, and the quality of the solution schedule will be evaluated, the last by comparison to a schedule manually derived by the Training Officer at Patrol Wing Ten (PATWING TEN).

Chapter II gives a summary of the development of scientific methods in scheduling. The scheduling function is also discussed.

Chapter III looks at the Training Officer's responsibilities. The resources used by the training officer and in the research problem are described. Chapter IV discusses the design and implementation of the prototype beginning with building the knowledge base. Heuristics, search methods, and optimization techniques are also discussed.

Chapter V offers a summary of the results obtained with the prototype. Comparisons will be made between two solutions. These will also be compared to the

manual solution. Chapter VI contains my recommendations for future work and some conclusions.

Appendix A is the source code for the prototype. Appropriate comments are made for documentation. Appendices B and C present demonstrations of runs with two different levels of optimization.

II. SCHEDULING

Since the early 1900's, numerous scientific scheduling techniques have been introduced. For example, Frederick W. Taylor developed the first scientific management techniques. He intended to improve the production cycle in the industrial environment. During World War I, Harry L. Gantt developed the Gantt chart for use in production scheduling. His chart became the popular bar graph. Simplicity and ease of application make the Gantt production chart a continued favorite for illustrating time-scaled problems. Some of these techniques had their roots in military application. Logistics support to American armed forces required sophisticated scheduling and management of resources.

The 1950's brought a tool for improved scheduling in the computer. The computer's advantages were obvious, but before this new capability could be tapped the schedulers and programmers had to mesh their knowledge. They had to effectively fashion logical methods of expressing scheduling approaches. Their efforts resulted in several advanced techniques. These include the Critical Path Method (CPM), PERT, and simulation. [Ref. 3]

A. SCHEDULING FUNCTION AND THEORY

1. The Scheduling Function

Allocating resources over time to accomplish a group of tasks (scheduling them) is a common practical problem. Scheduling presumes the tasks have been outlined and the resources available have been determined. In practice this definition is too simple; repeated bargaining can place between the planner and the scheduler. Each new proposal of the schedule may shed light on problems that were previously masked. The

resource availability may be modified as well as the tasks. This might be repeated several times until a satisfactory, and hopefully, optimum schedule is produced.

There is a rational method of deciding which scheduling strategy is best, which consists of four primary steps. In the first step, a subtle and often complicated process takes place: the problem is identified and the factors that guide the decision-making are formulated. The second step is the analysis of the elements of the problem and their interrelationships. The decision variables that are identified must follow specific relationships and constraints. Thirdly, the qualities of the feasible alternatives are examined. Finally, one of several scheduling algorithms that appear to be of use is selected. It meets the criteria initially established in the first step. [Ref. 2]

Pictorial representations can provide valuable help in the scheduling function. Even simplified graphs can represent the general structure and properties of problems. The problem discussed in detail in Chapters III and IV can be viewed as a bar graph: the time resource of each squadron is represented along a horizontal line.

2. Scheduling Theory

Mathematical models that relate to scheduling are a concern of scheduling theory. Scheduling theory first translates the goals of the problem into definitive objectives. Quantitative constraints are made of the decision-making restrictions. The problem is ultimately stated in concise mathematical form.

Scheduling theory considers three types of decision-making goals: utilization of resources, response to demands, and conformance to deadlines. A scheduling problem is solved by answering two basic questions: Which resource is allocated to handle each task? And what time will the task be performed? These are referred to as allocation and sequencing, respectively.

Resources are defined by their qualitative and quantitative capabilities. A task is characterized by its resource demand, its duration, the time frame in which it may be scheduled, and, on occasion, by precedence restrictions. The problem may contain one resource or many. Multiple resource problems usually include multistage tasks. When more that one project is involved the complexity of the problem rapidly increases to a degree that computerized assistance is beneficial. [Refs. 2, 4]

B. CHARACTERISTICS OF A SCHEDULER

O'Brien [Ref. 1] describes attributes a person should possess to be a good scheduler. They should have a keen organizational capability. Academic background for the field to be scheduled is warranted, and appropriate work experience helps. Knowledge of scheduling techniques and experience in applying them helps. In addition, good judgement in the selection of the basic assumptions for forecasting is needed. Care must be taken not to be overcommitted to one particular technique. Narrow-mindedness can hide or mask the uncertainties in the assumptions which must eventually be made.

C. SCHEDULING CATEGORIES

Scheduling techniques can be divided into four categories: time scheduling, resource scheduling, production scheduling, and general scheduling. The time scheduling techniques are based upon network logical plans which build on event start or finish times and assume the required resources are available. CPM, PERT, and precedence diagrams are examples. Most resource scheduling systems require more computation effort than the basic time scheduling; resource scheduling uses time scheduling as a foundation and builds the schedule around limited resources. Optimizing available resources and inventory control are examples of resource techniques. Production scheduling techniques are generally information systems that can be represented in the form of a graph. Both time and resource scheduling techniques can be

found in production scheduling. Production-oriented techniques have limited application beyond a production setting. General scheduling techniques, as the name suggests, are useful across time, resource, and production areas; the Gantt chart previously discussed is an example. General scheduling techniques covers a wide range of approaches, either to directly schedule or to amplify the results of other scheduling techniques. My research combines time and resource scheduling techniques.

D. SCHEDULING SOFTWARE

Computerized scheduling programs often employ search methods used in artificial intelligence (AI). Examples of the AI strategies are: the A-star (A*) algorithm, the B-star (B*) algorithm, bidirectional search, branch-and-bound algorithms, dependency-directed backtracking, and depth-first search. Each method has a different action selection procedure.

A difficulty with trying to find suitable software for scheduling is that much is task-specific. Davis [Ref. 5] observes:

> Although a person can do something with almost every problem, he will do terribly on almost every problem, except the ones he truly understands. A person who is not familiar with a problem can get somewhere but not very far. In fact, people solve problems well only when they know a great deal about the problem domain.

The system must have access to pertinent information. Formalizing knowledge and implementing knowledge bases are major tasks in the construction of systems that try to mimic how a person reasons. Correctly representing the problem domain is critical to efficient solution methods. A large system may require hundreds of rules and thousands of facts. These can be obtained through interviews and correspondence which can be tedious and time consuming. However, once created they are virtually immortal and readily accessible.

PATWING TEN has not bought an off-the-shelf PC program to solve their scheduling problems because the task requires handling of multiple squadron schedules. Available commercial software does not handle multiple projects and multiple resources simultaneously. Critical-path software exists for scheduling when the actions are know, precedences among actions are known, and durations of the actions are known. But critical-path software minimizes *total-time*, and does not provide niceness of gaps in schedules as does my program.

As already suggested, scheduling usually has a very large domain of possible answers. Simple algorithms soon encounter a combinatorial explosion that can exhaust the capacities of even large computers. Most real-world scheduling is NP-complete; hence, algorithms are $O(n^n)$ where n is the number of schedulings that are necessary to perform in the problem. [Refs. 6, 7]

III. THE TRAINING PROBLEM

A. PATROL WING TRAINING OFFICER

The Training Officer is responsible for the development of the operational readiness[1] of the squadrons. In pursuit of this goal, he must coordinate with each squadron's Training Officer, personnel within his own command, and the respective commands of inspection teams. Table 3.1 provides an excerpt from a PATWING TEN instruction which formally describes the Training Officer's responsibilities. Similar responsibilities exist for each Patrol Air Wing Training Officer. There is a large amount of real responsibility with this job; almost all facets of the squadron's ability to function are affected by decisions made by the Training Officer. Thorough preparations must be made to ensure that the job gets done right the first time.

B. RESOURCE DIRECTIVES

Regulations mandate requirements with little guidance for managing the resources. The resources in PATWING TEN's scheduling problem are time and inspection teams [Ref. 8]. If all the resources were controlled by the Training Officer the job would not be as complicated. But they are not. Time is consumed by events external to the Training Officer's control. The inspection team availability is influenced by demands other than that of the PATWING TEN Training Officer. Commander Patrol Wings Pacific (CPWP) Instruction C3500.24 outlines inspection requirements and all required

---

[1]The degree to which a military unit is capable of performing its primary roles.

evaluations a squadron must complete during its at-home[2] period. Specific inspection and periodicity requirements are presented in other references. The submittal of the yearly training plan is dependent upon adequate coordination with all evaluating commands regardless of their location.

Information regarding the Nuclear Training Proficiency Inspection (NTPI) can be found in Commander Naval Air Pacific (CNAP) Instruction C8121.1. The role of chief inspector for PATWING TWO and PATWING TEN is delegated to Commander Patrol Wing Two (COMPATWING TWO), Naval Air Station, Barbers Point, Hawaii, and COMPATWING TEN respectively. The inspection is technically conducted by a team from Commander Nuclear Weapons Training Group Pacific, San Diego, California. This command conducts NTPIs and similar types of inspections for all nuclear-capable units of the Pacific region extending as far as the Indian Ocean. Committing this one inspection team to a squadron NTPI is difficult since competition is always high. Due to the high demand, actual commitment to an inspection date is made quarterly. However, as with all the inspections, the periodicity requirement limits the latest the NTPI can be scheduled, and the annual Training Plan reflects this date.

The Mine Readiness Certification Inspection (MRCI) requirements are contained in Chief of Naval Operations (OPNAV) Instruction C5040.15C. Unlike the NTPI, which has teams available through commands on the east and west coast, the MRCI is conducted solely by a team from the Commanding Officer, Mine Warfare Inspection Group, Charleston, South Carolina. They are responsible for the MRCI and similar inspections on all commands of mine-warfare capability. Again competition for this resource is high but commitment can be obtained on an annual basis.

---

[2]The period when the squadron is not deployed or otherwise detached.

Table 3.1 TRAINING OFFICER RESPONSIBILITIES

COMPACTWINGTENSTAFFINST 5400.1A
MAY 13, 1986

SECTION 4

TRAINING DEPARTMENT

2401. *Training Officer(50).* The Training Officer is responsible to the Assistant Chief Staff Officer for Readiness for all matters pertaining to tactical training and for developing and maintaining the maximum degree of operational readiness of assigned units. In the fulfillment of these responsibilities the Training Officer shall:

a. Prepare all PATWING TEN programs and directives for the training of assigned units in the areas of ASW, weapons, navigation, maritime surveillance, and pilot, NFO, and aircrew proficiency.

b. Monitor the overall training programs for PATWING TEN squadrons and make recommendations to respective squadrons as to action necessary to correct any noted discrepancies.

c. Monitor the ASW readiness of all assigned units. Analyze PATWING TEN readiness data and evaluate unit readiness trends. Provide guidance to respective squadrons as necessary to improve readiness.

d. Develop training requirements and coordinate with the Operations Officer in scheduling of services necessary to satisfy these requirements.

e. Maintain close liaison with COMPATWINGSPAC Training and Readiness Officers and squadron Training Officers.

f. Coordinate with the Current Operations Officer in the scheduling of training flights.

g. Ensure conduct of weapons technical training and inspections, and maintain the weapons readiness of PATWING TEN squadrons.

CNAP Instruction 8023.3 provides information regarding the Conventional Weapons Technical Proficiency Inspection (CTPI). The team conducting this inspection is internal to Commander, Patrol Wings Pacific, NAS Moffett Field, California,

(COMPATWINGSPAC). The proximity of the CTPI inspection team eases the communications gap allowing frequent conferences and facilitating the planning function required prior to drafting the schedule.

The Patrol Wings U.S. Pacific Fleet Command Inspection Program (CI) is an administrative inspection outlined in CPWP Instruction 5040.3A. Generally, all Naval units must undergo a similar inspection. Like the CTPI, it is lead by a local team. The same advantages apply for this locality as do the same precautions.

Relevant direction for the Naval Air Training and Operating Procedures Standardization Program (NATOPS) can be found in OPNAV Instruction 3710.7L. NATOPS is another kind of inspection, designed to improve combat readiness and achieve a substantial reduction in the aircraft accident rate. There is one NATOPS inspection team for COMPATWINGPAC. They evaluate squadrons for both COMPATWING TWO and COMPATWING TEN. The team is also based at Naval Air Station Moffett Field.

Both the NTPI and the MRCI have pre-inspections conducted internal to each squadron (i.e., pre-NTPI and pre-MRCI). They are at least equal in duration to the primary inspections and are preparatory. While they do not demand a particular inspection team resource, they do consume time.

The Operational Readiness Evaluation (ORE) is a series of exercises used to determine the overall operational readiness of a squadron. The Patrol Aviation Qualifications Exercise Manual, CPWP Instruction 3500.2b, outlines each exercise and required proficiency. COMPATWING TEN is delegated with conducting the evaluations which extend over several weeks. The squadron is vulnerable to scheduled ORE events at almost any time in this period. Later discussion will describe events which are allowed to encroach on this time.

The key scheduling issue is competition for inspection team availability [Ref. 9]. This necessitates coordination not only with the commands providing these resources but also with the PATWING TWO Training Officer. PATWING TWO's squadrons demand sharable resources at a level commensurate with that of PATWING TEN. Frequent conferences, formal and informal, are necessary to resolve most of the conflicts.

The PATWING TEN Maintenance Officer can report on aircraft availability. Squadrons periodically change to a different type of aircraft. More frequently, a squadron loses an aircraft to the Naval Air Rework Facility (NARF) for an overhaul. If a change is in progress, the squadron is severely restricted in the type of evaluations which can be held; retraining of all the crew members must take place. Certain changes invalidate previous inspections. To accommodate this the at-home period may be extended. But whatever the reason for aircraft unavailability, it can diminish scheduling opportunities.

C. TYPICAL SQUADRON TRAINING CYCLE

The life cycle of a patrol squadron has two major components: the deployment and the at-home period. For a deployment, the squadron is transplanted to a remote land-based airfield for six months. Operational tempo precludes a structured training environment upon which an inspection schedule can be built. Obviously, the squadron is operationally ready or it would not be deployed. Therefore, the Training Officer does not normally schedule a squadron for any inspections during this time.

When the deployment ends the squadron returns to its home base. Normally, this is for eleven months. Factors which can affect the length of this period include aircraft type changes, change of deployment sites, funding, and world politics. From this eleven months the Training Officer must find time to ensure their requalification prior to the next deployment. Eleven months may seem like ample time to schedule the necessary

inspections and training events, but this period is full of activity and other squadrons frequently compete for the same resources. The squadron must compete for the time to conduct their daily business in addition to the events scheduled by the Training Officer.

The first 30 days back from deployment is called the "post-deployment safety stand-down." Many personnel transfers are delayed until this time; air crew integrity is weakened; squadron members go on leave; families are rejoined. These adjustments detract from a training environment. The Training Officer does not schedule the squadron during the post-deployment safety stand-down.

After the post-deployment safety stand-down, the squadron has probably received the majority of its personnel replacements. Crew integrity is reforming and safe training can take place. Because of the periodicity requirements of the NTPI, the pre-NTPI is often the first major training event and preparations begin immediately. Other training may run concurrently, but the preparedness for the NTPI is foremost. The pre-NTPI attempts to duplicate the strict conditions the NTPI team imposes. Of course, the NTPI date must be known before the pre-NTPI is scheduled, to ensure an adequate lead time of two to three weeks. The NTPI lasts two working days. The squadron is inspected to ensure compliance with the regulations set forth by the various nuclear regulatory agencies. In regard to a patrol squadron's nuclear weapon capability, everything from health records to the weapon itself is carefully scrutinized. The Training Officer makes every attempt not to schedule anything else during the period of the pre-NTPI and NTPI, which is up to twenty-six days.

The next major event could be the ready alert. It requires an aircraft and crew be prepared for short-notice take-off twenty-four hours a day. The tasking is normally for a full month with limited exceptions. The squadron handles this by rotating crews daily. At least one backup crew and aircraft is provided; the backup crew assumes the ready

alert status if the primary crew is sent on a mission. This is extremely demanding on personnel resources: both crews must be properly rested, limiting their participation in other workday activities, and the Training Officer cannot use this time. The ready-alert tasking normally occurs twice during the at-home period and is never consecutive. On occasion, a forty-five day ready alert may be tasked due to the unavailability of a squadron.

The NATOPS inspection usually occurs either before the squadron's first ready alert or between the two ready alerts. This is a comprehensive evaluation of the aircrew's knowledge about the aircraft. The crews are tested through written exams and actual flights. Aircraft systems, performance, and emergencies are included. There are eight types of crew members for each aircraft; each is independently evaluated by inspection team specialists. The underlying emphasis of NATOPS is safety. An aircrewman that does not pass the NATOPS is grounded[3] until a passing grade is made. The squadron itself receives an overall grade; if this is too low the squadron as a whole can be grounded. This is the longest of the inspections, taking up to ten working days. The Training Officer must allocate an uninterrupted two-week period for the NATOPS inspection.

A three month time frame is dedicated to ORE vulnerability. This ends about forty-five days prior to the squadron deploying. Within this period the CTPI, pre-MRCI, MRCI and CI are usually scheduled. The squadron has usually just completed the second ready alert when this period begins.

The CTPI and pre-MRCI usually occur simultaneously. They are both three days in duration, early in the first month. Similar to the pre-NTPI and NTPI, a block is preserved

---

[3] Not permitted to actively participate in the mission during flight.

for the pre-MRCI and MRCI. The MRCI follows two to three weeks after the pre-MRCI. Almost every capable aircraft joins in this mining exercise. Mining is a primary role of a patrol squadron and satisfactory performance is mandatory before deployment.

It may be that the ready alert extends up to fifteen days into the ORE period. If this is the case, the CTPI/pre-MRCI is delayed until after the ready alert. Following the MRCI, the squadron is usually heavily involved in the ORE activities. These include intelligence briefings, simulated operational scenarios, and recognition of Soviet-bloc vessels. It is to the ORE that the squadron dedicates its energy in final preparation for deployment.

The last inspection held prior to the deployment is the Command Inspection. It is one day, preferably the last working day of the ORE vulnerability period. This determines if the squadron is administratively fit for deployment. An examination is made of squadron instructions, office files, and administrative procedures.

The forty-five days prior to deployment are not usually committed to any major events and are held in reserve. If there are problems in the ORE vulnerability period or an early deployment is necessary, that time is available. Large-scale rescheduling is then normally not necessary if either of these happen. This time block consumes approximately fourteen percent of the initial eleven month at-home period.

IV. DESIGN AND IMPLEMENTATION

The prototype program was tested on ISI workstations under the UNIX operating system and was written in an interpreter language C-Prolog. My program produces a PATWING TEN Training Plan for fiscal year 1986. For de-classification purposes, deployment sites are not named. Appendices A, B, C contain the source code used to implement the scheduler. Appendices B and C are the search programs provided by Prof. Neil C. Rowe at the Naval Postgraduate School. [Ref. 7]

A. PROTOTYPE ORGANIZATION

My prototype program is written in the language Prolog. It has 13 modules:

- **scheduler**
- **database**
- **generator**
- **depthsearch**
- **depthfirst complement**
- **estimator**
- **nopath search**
- **nopath search complement**
- **cost**
- **earmark**
- **calendar**
- **schedule-writer**
- **utilities**

The **scheduler** contains the primary driver and maintenance routines. The driver is automatically initiated when **scheduler** is loaded into a Prolog interpreter. This causes the **database**, **generator**, **depthfirst**, **calendar**, and **utilities** modules to be loaded. **Scheduler** stores output in the file **FISCAL-YEAR-1986**, which also establishes a

database for the next year. As the program is executing, the scheduler times each process and computes the average cpu time needed for each immediate branch that occurs during the search for the solution. A branch is the addition of one scheduling assignment to a list of previously-made assingments. These branches are also called transition states or successors; the terms often used interchangeably.

The **database** module has two purposes: it provides the initial data necessary to start scheduling and converts the prerequisite dates into structures that are more efficiently manipulated. **Generator** contains preprocessing rules that compute every possible period in which an event might be scheduled for the fiscal year. Fiscal year 1986 has 660 of these which the program refers to as **trialperiods**.

Depthsearch (Appendix B), activated through predicate **search2**, determines the ready alerts for PATWING TEN squadrons, the first of three subsearches. The **depthfirst complement** defines the rules needed for thi

One difficulty with a dynamic search problem like this is determining the number of events that will be scheduled for the fiscal year concerned. **Estimator** makes a best guess by estimating how many events can be scheduled under certain conditions. These guesses are asserted as facts (e.g., **goalsubtotal1(16)** and **goalsubtotal2(15)**).

Appendix C contains a search algorithm similar to **A*** search, but modified for this research by Prof. Rowe; I used this **nopathsearch** in my prototype. **Nopathsearch** uses the **nopathsearch complement** and **cost** modules to conduct the second and third subsearches and complete the scheduling. The code in **cost** computes the relative value of each successor state as the search progresses. Any event which can be scheduled after the current list of events included in the successor state. Once the third subsearch is complete, the **scheduler** module puts flags into the database marking the latest possible date an event may be scheduled the next fiscal year. These flags will be used in

conjunction with the solution to initialize the program when the following year's schedule is drafted.

The **schedule-writer** module provides a simple output of the schedule in the form of a list in chronological order. The **utilities** module includes some basic list-manipulating predicates that are used throughout the program.

Some assumptions are made in my system:

- The database contains valid dates in the correct date format.
- The ready-alert requirements will not change.
- The periodicity regulations for training will not change.
- No long term accelerated operational tempo will occur.
- Maintenance requirements are not relevant.
- The information obtained through the interviews with LCDR George Sanford[4] is translated correctly to the program.

B. COLLECTING THE KNOWLEDGE BASE

Chapter III pointed out that this scheduling problem is quite knowledge-specific. Information regarding the existing scheduling procedures at PATWING TEN had to be obtained to correctly build the knowledge base for the scheduler. This was achieved through two interviews with LCDR Sanford, though the author's own experience in the Patrol community helped. While this prototype is not an expert system, LCDR Sanford delivered the necessary information with the clarity and preciseness of an expert.

The first interview provided a procedural description of preparing the training plan. A similar account is available in an informal turnover notebook the Training Officer prepares for his relief. The PATWING TEN training plan has always been manually developed [Ref. 9] following this algorithm:

---

[4] PATWING TEN Training Officer 1985-1988.

1. Fill in the deployment periods for each squadron. A deployment begins and ends on the tenth of a month.

2. Compute the latest date each inspection can take place in accordance with the periodicity requirements. Mark these on the schedule draft.

3. For each squadron, annotate the draft with the following periods:

a. Its ORE vulnerability period.

b. 45 days prior to its deployment.

c. Post-deployment safety stand-down.

4. Assign the ready alerts.

a. Determine which squadron has the ready alert the last month of the current planning year. This is the starting point for future ready-alerts.

b. When selecting a ready-alert do not consider any squadron that:

(1) Held the ready alert the month prior to the month being scheduled.

(2) Is on deployment.

(3) Is in its post-deployment safety stand-down.

(4) Is in the ORE vulnerability period.

(5) Is the the period 45 days prior to deployment.

c. If more than one squadron remains from part b: select one that has not yet been scheduled for a ready alert during its current at-home period.

d. If no squadron is available for part c: temporarily skip to the next month and select a squadron in accordance with parts b and c. Then split the skipped month between the previous ready alert and the selected ready alert squadron. This will assign the ready alert to two squadrons over a three month period.

e. Consider the following as the preferred order to select from available squadrons:

(1) Those in third full month or later following post-deployment safety stand-down.

(2) Those in second full month or later after the post-deployment safety stand-down.

(3) Those in first month of ORE vulnerability period.

f. Continue steps b through e until some squadron is assigned a ready-alert for every month.

5. Schedule the NTPI for each squadron (as required):

a. The date must be after the safety stand-down and not during any ready alert.

b. Prefer the latest date possible.

c. There must be prior time for the pre-NTPI.

d. Ensure that no major holiday[5] interrupts the pre-NTPI/NTPI block.

6. Schedule each pre-NTPI. The date must fall after the safety stand-down.

7. Using the due dates marked on the draft, schedule the rest of the activities. None should be during a ready alert and all should be after the post-deployment safety stand-down.

a. The NATOPS evaluation should be as early as practical.

b. The CTPI/pre-MRCI should be as early in the ORE vulnerability period as practical.

c. The MRCI should be no earlier than two weeks after the pre-MRCI, with the same restrictions as the pre-NTPI/NTPI.

d. The CI should be on the last working day of the ORE vulnerability period.

8. Make necessary adjustments to optimize the overall schedule and the resources with which it is built.

The last step in the algorithm is the most difficult, and requires an experienced scheduler [Ref. 9]. It involves distributing the inspection activity as evenly as possible through the squadron's at-home period and among the respective inspection teams.

---

[5] A major holiday is a federal holiday that encourages more than one day of leave for a significant number of command personnel.

The interview lasted approximately 45 minutes. LCDR Sanford was asked to later review the information we covered for possible exceptions. Additional information was not received regarding exceptions prior to translating information obtained from the interview into a program.

A second interview was conducted for verification once the prototype was functioning. In this interview, I described how the prototype determined a schedule. Using the "if,then" style of prolog, I presented the decision logic of the prototype scheduler. LCDR Sanford was to interrupt only when an incorrect or out-of-sequence step occurred. Only minor refinements were recommended.

C. SEARCH METHODS

In all, three searches are conducted using two search algorithms. Initially I attempted a solution to this problem with one search. It became evident that the number of solution paths explored, even at an early phase, was large and the answer would require much processing time. Closer examination of the problem suggested decomposition into two subsearches [Ref. 7] with the ready alerts scheduled in the first search. Even though this division was helpful, there was still a lengthy second search. The problem was then further decomposed by scheduling the pre-NTPI, NTPI and NATOPS in one search process and the CTPI/pre-MRCI, MRCI, and CI in another; these inspections can be put in either the ORE vulnerability period or not. Once all three searches are solved, and the next due date for each inspection is tagged, the three search answers are appended for input to **schedule-writer.**

Two different search methods are now used with the three subsearches. **Depthsearch** (depth-first search) is used to schedule the ready alerts and **nopathsearch** the inspections. Both strategies call successor rules which give state transitions. These rules modularize the various parts of the Training Officer's algorithm. The state

transition is a possible next event which can be scheduled with the events provided as input to the successor rule. Both methods also require a **goalreached** rule defining when the search succeeds.

A depth-first search is done for the ready alerts since the number of suitable assignments of them is low and any suitable assignment is satisfactory. Prof. Rowe [Ref. 7] provided the depthfirst driver in Appendix B. One addition is made to Rowe's depth-first search to count the number of successors generated during the search. During the following two searches repeated references are made to the ready alerts. The list processing is extensive and caching reduces it.

Appendix C illustrates Prof. Rowe's **nopathsearch** which is a modified A* search strategy. A* search is a form of heuristic search that tries to find the cheapest path from an initial state to a goal state [Ref. 7]. Nopath search differs from A* search in two ways. As the name suggests, **nopathsearch** does not keep path lists found in trying to reach the goal. Instead, an agenda holds the currently considered transition states as facts. Secondly, a pruning function keeps the agenda size down. The pruning function inspects the agenda removing items that are permutations of the current best state and also items whose estimated value is K (a parameter) worse than that of the best state; the larger this K, the more items are left on the agenda and the longer it takes to find a solution. I expanded the **nopathsearch** with utility predicates for counting the successors, keeping track of the agenda size before and after pruning, and writing the best state picked (see Appendix C).

My program puts a cost on how far removed a state is from the proposed optimum of 30 days between inspections. When computing the cost of a state two relationships are examined: The number of days between inspections within each squadron (**squadroncost**) and the number of days between inspections conducted by the same

inspection team (**teamcost**). In both cases thirty days is assumed as the optimum period. Cost should increase sooner for gaps less than thirty days than those greater than thirty. Prof. Rowe's formula for calculating this cost is

$$\text{Cost is } \exp((24-D)/12) + \exp((D-24)/30) .$$

The **squadroncost** is the cost of waiting to schedule an inspection beyond the first of the fiscal year:

$$\text{Cost is } \exp((D-24)/30) .$$

The **squadroncosts** and **teamcosts** are summed for the overall cost to that state.

The evaluation function needed for A* search reflects how far the state is from the goal by multiplying the number of events left to schedule by cost's lower bound of 1.82793. An estimate is prepared by the rule **goalstatetotal** as to how many events will be scheduled for that fiscal year; the training plan varies in the number of events scheduled due to the staggered effect of the at-home periods. The estimate is made by computing how many of the required inspections could possibly be scheduled in the available time each squadron has for the year. The search goal is reached when the number of events in the current state is equal to **goalstatetotal's** estimate.

D. TIME MANIPULATION

The proper management of dates is critical to the program. The code in **calendar** manipulates dates by computing starts and finishes, determining interference, and calculating windows of opportunity. The conversion of the event dates necessary to initialize the program to integer dates and creation of the **trialperiods** exercise **calendar** the most; it is not required again until the solution is generated by **schedule-writer.** It is designed to work for any given or calculated date from 1 January 1600.

Functions performed by **calendar** include:

- date to integer
- integer to date
- day of week
- difference between dates
- date calculation by adding or subtracting days
- holiday
- leapyear

The required format for dates is [Day,Month,Year] as in the Gregorian calendar. The Gregorian calendar modifies the Julian to lose three days every 400 years. Leapyear assists in this adjustment by adding one day to the normal 365 day year every four years except on a centennial year. An exception to this rule is that a centennial year is also a leapyear if it is divisible by 400.

The rules are flexible enough to function with integers or Gregorian dates. **Calendar** treats dates as integers unless some instance must be checked for being a holiday or particular day of the week. Holidays must be avoided when scheduling inspection teams prevent an unwanted interruption in the inspection.

E. DATA STRUCTURES

Table 4.1 defines the event symbols used in the program. The events are of three types: evaluative, tasking, and deployment. The symbol ds0 means a deployment in which the fiscal year begins. The at-home period for a squadron after ds0 is termed period 1. Period 2 for a squadron starts after the safety stand-down of a deployment that begins after the fiscal year. These terms are used occasionally to distinguish to which at-home cycle is being referred.

The **scheduler** takes the predetermined periods contained in the file **database** and asserts them as **prerequisite** and **priorevent** facts. These periods include all deployment-related events and events scheduled in the previous fiscal year. For instance,

Table 4.1 EVENT SYMBOLOGY

SYMBOL	EVENT TYPE	EVENT NAME
ewp	nuclear weapon, evaluative	pre-NTPI
ewn	nuclear weapon, evaluative	NTPI
en	natops, evaluative	NATOPS
ewc	conventional weapon, evaluative	CTPI
ewm	mining weapon, evaluative	MRCI
ec	command, evaluative	CI
tr1a	ready alert, tasking	Ready Alert, 1st assignment
tr1b	ready alert, tasking	Ready Alert, 2nd assignment
dv	vulnerable, deployment	ORE vulnerability period
dr	operational, deployment	Deployment period
ds0	post stand-down, deployment	Safety stand-down
ds1	post stand-down, deployment	Safety stand-down

Patrol Squadron Nine's post-deployment safety stand down

[vp9,ds0,[11,dec,1985],[10,jan,1986]]

is converted and asserted as

prerequisite(vp9,ds0,140963,140994).

If Patrol Squadron Nineteen had the last ready alert in the previous fiscal year, this is asserted as

priorevent(vp19,tr1a,140863,140908).

The input database also contains facts about the last possible date by which each inspection must be completed:

earmark(vp47,ewm,[15,oct,1985]).

The Training Officer uses a similar flag when he initially sets up his scheduling chart by marking the latest date each inspection can be scheduled.

Before the search routines are initiated all the possible start and finish dates are calculated for each event and asserted as **trialperiod** facts.

trailperiod(ewc,140173,140175).

This saves time and the data generated can be cached; caching allows separate runs to be made for different combinations of squadrons without having to wait for preprocessing. During calculation of start and finish dates, precautions are taken to preclude periods from being interrupted by federal holidays or weekends. Each event type has its own rules primarily because its duration is different. The ready-alert **trialperiods** are unrestricted. The NTPI trialperiods are two days long and can begin and end on all days but holidays and weekends. Two weeks is allowed for the NATOPS inspection, a period starting on a Monday followed by ten working days ending on a Friday. The CTPI and MRCI trialperiods are similarly computed but are only five days long.

Each of the three different subsearches has its own of successor rules; the predicate names are **successor1**, **successor2**, and **successor3**. The rules right sides define the constraints that must be met for the successor to succeed and produce a state transition in the form of a list, the second argument to each successor predicate of my program. For example,

successor1(CurrentState,[event(vp9,tr1a,Start,Finish)|CurrentState])

is the predicate used by one of **search1's** successor rules. Each branch of the search is appended as an event to the list of scheduled events that led to the branch. The event predicate has four arguments:

event(Squadron,EventName,Start,Finish).

The squadron is one of those in **squadronlist**:

squadronlist([vp9,vp19,vp40,vp47,vp48,vp50]).

The name of the event must belong to the list in **eventnames**:

eventnames([tr1a,tr1b,ewp,ewn,en,ewc,ewm,ec]).

F. THE SEARCHES

Three searches are used to find three components of the final solution: the ready alert assignments, the non-ORE inspection schedule, and the ORE-related inspection dates. By decomposing the problem into these three searches an answer is found quicker and more efficiently.

1. First Search

This search provides ready-alert tasking for the entire fiscal year. The **scheduler** proceeds to call search1, which in turn calls **depthsearch**. The goal, the ready alert schedule, is satisfied through a series of applications of **successor1** rules. When **search1** begins, it looks at the first **successor1** rule and tries to satisfy the constraints on the right hand side. If it cannot succeed it calls the next **successor1**. Upon succeeding a ready alert is assigned. If no rule succeeds, **depthsearch** fails, and the program terminates informing the user an answer cannot be determined.

This first search has 17 **successor1** rules. Each has specific heuristics to guide the search. There are three rules to schedule the first ready alert of the year, ten rules to schedule ready-alerts for the other squadrons, and four rules for possible second ready-alert assignments. The subordinate rule **pick_nextsquadron** determines which squadron gets the assignment. As with the manual implementation the rule requires a look-up of which squadron last had the ready alert; this is done by finding the **priorevent** of event name "trla" or "trlb" having the latest date.

Once each squadron has been one ready-alert period, **successor1** uses similar preferences in assigning the second ready-alerts if necessary. The last two **successor1** rules provide the means to split three months between two squadrons for month in which no squadron qualifies. The month is shared by extending the previous squadron's period

and beginning the one's early. This list of events is asserted as **readyevents** for easy access by **search2** and **search3**.

2. Second Search

The second search uses the **nopathsearch** strategy. This determines which is the best next state according to the sum of the **cost** and **eval1** functions. There are ten **successor2** rules, ordered by priority. Within these rules the squadron names are sequentially selected from the **squadronlist** and tested for qualification for event assignment. In scheduling the pre-NTPI, NTPI, or NATOPS inspections, **successor2** rules make the following assumptions:

- These inspections have not been previously scheduled.
- A post-deployment safety-standdown has been scheduled.
- A trialperiod for this event has been scheduled.
- The trialperiod is after the safety-standdown.
- The trialperiod is before the event due date (i.e., **earmark**).
- The trialperiod is before the ORE-vulnerability period.
- The trialperiod does not interfere with any previously scheduled event for that squadron.
- The trialperiod does not interfere with any previously scheduled event for the inspection team for that event.

In addition, specific heuristics prevent events from being in other undesirable periods; these can be modified by the programmer. An event cannot start until a certain number of days before or after other scheduled events or **prerequisite** periods. Other constraints prevent events with the same inspection teams and events for the same squadron from overlapping or interfering; relaxing these constraints is permissible. This should be done only after careful justification and consideration of the consequences.

3. Third Search

Inspection-team assignments required a third search called **search3**; the same **nopath** strategy is applied as in **search2**. **Search3** schedules the ORE-related events

(i.e., CTPI/pre-MRCI, MRCI, and CI). **Search3** differs from **search2** in that:

- The trialperiod is during the ORE vulnerability period.
- The CTPI/pre-MRCI is the only inspection that competes with a ready alert for time allotment.

With only four **successor3** rules, **search3** does not have to contend with many alternatives. The CTPI/pre-MRCI has two rules; the difference between them is the **readypresent** subordinate rule. When **readypresent** succeeds the **trialperiod** fails until unless it is after the ready alert.

G. FINAL SOLUTION

Once all three searches have succeeded, the goal states of **search2** and **search3** are concatenated. This list is forwarded to **earmark_dropdeaddates** which flags the next due date for each of the primary inspections. The old **earmark** fact is removed and the new ones asserted. These flags are for the next fiscal year.

Lastly, the **scheduler** calls **schedule-writer** which sorts the results from **search1**, **search2**, and **search3**. This list is printed chronologically by start dates and is made part of the database for next year's use along with the **earmark** facts. Figure 4.1 is an example of the output the prototype produces.

TRAINING SCHEDULE FOR FISCAL YEAR 1986

```
event(vp47,ewm,[1,oct,1985],[4,oct,1985])
event(vp46,ewn,[1,oct,1985],[2,oct,1985])
event(vp48,ec,[7,oct,1985],[11,oct,1985])
event(vp46,tr1a,[16,oct,1985],[30,nov,1985])
event(vp19,ec,[21,oct,1985],[25,oct,1985])
event(vp40,ewp,[31,oct,1985],[1,nov,1985])
event(vp47,ec,[18,nov,1985],[22,nov,1985])
event(vp40,ewn,[18,nov,1985],[19,nov,1985])
event(vp40,tr1a,[1,dec,1985],[31,dec,1985])
event(vp46,en,[2,dec,1985],[13,dec,1985])
event(vp46,tr1b,[1,jan,1986],[31,jan,1986])
event(vp40,en,[6,jan,1986],[17,jan,1986])
event(vp40,tr1b,[1,feb,1986],[28,feb,1986])
event(vp9,ewp,[6,feb,1986],[7,feb,1986])
event(vp9,ewn,[24,feb,1986],[25,feb,1986])
event(vp9,tr1a,[1,mar,1986],[31,mar,1986])
event(vp50,ewp,[3,mar,1986],[4,mar,1986])
event(vp46,ewc,[4,mar,1986],[7,mar,1986])
event(vp50,ewn,[25,mar,1986],[26,mar,1986])
event(vp46,ewm,[31,mar,1986],[3,apr,1986])
event(vp40,ewc,[1,apr,1986],[4,apr,1986])
event(vp50,tr1a,[1,apr,1986],[15,may,1986])
event(vp9,en,[14,apr,1986],[25,apr.1986])
event(vp40,ewm,[28,apr,1986],[1,may,1986])
event(vp46,ec,[5,may,1986],[9,may,1986])
event(vp9,tr1b,[16,may,1986],[30,jun,1986])
event(vp40,ec,[2,jun,1986],[6,jun,1986])
event(vp50,en,[2,jun,1986],[13,jun,1986])
event(vp50,tr1b,[1,jul,1986],[15,aug,1986])
event(vp9,ewc,[8,jul,1986],[11,jul,1986])
event(vp48,en,[14,jul,1986],[25,jul,1986])
event(vp9,ewm,[5,aug,1986],[8,aug,1986])
event(vp47,en,[11,aug,1986],[22,aug,1986])
event(vp48,tr1a,[16,aug,1986],[30,sep,1986])
event(vp50,ewc,[2,sep,1986],[5,sep,1986])
event(vp47,ewp,[3,sep,1986],[4,sep,1986])
event(vp9,ec,[8,sep,1986],[12,sep,1986])
event(vp19,en,[15,sep,1986],[26,sep,1986])
event(vp47,ewn,[22,sep,1986],[23,sep,1986])
event(vp50,ewm,[29,sep,1986],[2,oct,1986])
```

Figure 4.1 Prototype Schedule Example

V. RESULTS

A. PROGRAM PERFORMANCE

The four main test runs are contained in Appendices D, E, F, and G. Theyu show the same problem solved four ways, the only differences being in the pruning variable K.

1. Storage Requirements

Test runs were conducted to study program performance. Three internal data areas used by C-Prolog were observed since the available storage among the areas is not automatically adjusted when a program exceeds the initial configuration. Table 5.1 shows that while global and local stack requirements remain fairly constant, heap requirements do increase when the scope of the search is broadened. The combined size of the twelve modules that make up the program is 54.6 Kilobytes.

2. Processing Time

Only one variable was changed for each test run. This was the pruning variable K that defines the agenda cost window in searches 2 and 3. The effect this had on processing time is illustrated by Table 5.2. Interestingly, the average time to process the successors is nearly constant.

The utility routines exhibited fairly constant performance as evidenced by Table 5.3. They consumed the 132 cpu seconds remaining of the final total in Table 5.2.

Table 5.1 SYSTEM REQUIREMENTS (K-BYTES)

Pruning Factor	Heap	Global Stack	Local Stack
K = 0	156.2	54.3	32.1
K = .5	165.98	55.03	32.3
K = 1.0	188.98	55.3	32.2

Table 5.2 PROCESSING SUMMARY

OPTIMIZING FACTOR	Search 1			Search 2			Search 3			Final		
	Succ	Avg	Total	Succ	Avg	Total	Succ	Avg	Total	Succ	Avg	Total
K = 0	19	.57	11	592	1.41	836	301	1.41	424	912	1.41	1403
K = .5	19	.59	11	979	1.69	1653	517	1.40	725	1515	1.66	2521
K = 1.0	19	.58	11	3047	1.61	4913	1040	1.45	1512	4106	1.59	6568
K = 1.5	19	.58	11	8498	1.87	15882	5735	1.19	6809	14252	1.58	22534

Succ = Number of Successors generated by this search.
Avg = Average processing time per successor (cpu seconds).
Total = Total processing time for each phase (cpu seconds).

Table 5.3 CONSTANTS IN PROCESSING

Process	CPU seconds
Converting the database	4.4
Generating 660 trial periods	117.567
Computing goal state	3.27
Flagging drop dead dates	0.77

3. Comparison with Manual Schedule

Table 5.4 compares the schedules produced by the test runs with a schedule made manually. The displacement of days from the date of the manual method is for each activity. Interestingly enough, the program scheduled events mostly earlier than the manual schedule and would, on occasion, schedule a seemingly less optimal schedule when the pruning variable was set at 1.5; some instances when the latter occurred are vp40 NATOPS, vp40 MRCI, vp47 CI, and vp50 NTPI. This could indicate a faulty cost function, but a more probable hypothesis is that when K = 0.5 the results most closely approximates the manual method and indeed when K = 1.5 the results provide a more optimal solution than the manual method considering the number of solution paths a person would have to mentally process to reproduce the results observed in the fourth test run. The optimization function is defined more clearly by comparing the cost of the final states. The manual method's final state has a cost of 675.677 while the answer when K =

Table 5.4 FISCAL YEAR 1986 COMPARISON

TRAINING SCHEDULE	Manual Method	K = 0	K = .5	K = 1.0	K = 1.5
Activity	Completed	Difference	Difference	Difference	Difference
vp9 pre-NTPI	20Feb86	-16	-13	-14	-13
vp9 NTPI	21Mar86	-30	-24	-24	-24
vp9 Ready-Alert	30Apr86	-30	-30	-30	-30
vp9 NATOPS	16May86	-14	-19	-19	-14
vp9 Ready-Alert	30Jun86	0	0	0	0
vp9 CTPI pre-MRCI	11Jul86	-1	0	0	0
vp9 MRCI	31Jul86	+7	+8	+8	+8
vp9 CI	30Sep86	-18	-18	-18	-18
vp19 NATOPS	26Sep86	0	0	0	0
vp19 CI	15Nov85	-20	-7	-7	-14
vp40 pre-NTPI	11Nov85	-20	-20	-13	-13
vp40 NTPI	21Nov85	+10	+7	+14	+14
vp40 Ready-Alert	-	NA	NA	NA	NA
vp40 NATOPS	7Feb86	-21	-21	-21	+35
vp40 Ready-Alert	31Mar86	-31	-31	-31	-31
vp40 CTPI pre-MRCI	11Apr86	-7	-7	-7	-7
vp40 MRCI	8May86	-7	0	0	+8
vp40 CI	27Jun86	-21	-21	-14	-21
vp46 NTPI	10Oct85	-8	-8	-8	-8
vp46 Ready-Alert	30Nov85	0	0	0	0
vp46 NATOPS	13Dec85	0	0	0	0
vp46 Ready-Alert	15Feb86	-15	-15	-15	-15
vp46 CTPI pre-MRCI	27Feb86	+7	+8	+8	+8
vp46 MRCI	27Mar86	+7	+7	+7	+7
vp46 CI	28May86	-19	-19	-19	-19
vp47 MRCI	27Sep85	+7	+7	+7	+7
vp47 CI	27Nov85	-5	-5	-5	-19
vp47 NATOPS	22Aug86	0	0	0	0
vp47 pre-NTPI	-	NA	NA	NA	NA
vp47 NTPI	-	NA	NA	NA	NA
vp48 CI	29Oct85	-17	-17	-17	-17
vp48 NATOPS	6Aug86	-11	-11	-11	-11
vp48 Ready-Alert	30Sep86	0	0	0	0
vp50 pre-NTPI	26Mar86	-22	-22	-22	-22
vp50 NTPI	25Apr86	-27	-29	-30	-35
vp50 Ready-Alert	31May86	-16	-16	-16	-16
vp50 NATOPS	7Mar86	+6	+6	+6	+6
vp50 Ready-Alert	15Aug86	0	0	0	0
vp50 CTPI pre-MRCI	5Sep86	0	0	-7	-7
vp50 MRCI	25Sep86	0	0	+7	+7

1.5 costs 403.528, which is a 40% improvement. Factors other than the pruning variable which influenced the differences with the manual schedule include the difficulty in mentally optimizing so many events; the fact that maintenance considerations in the manual method are not yet implemented in the program; and because of an aircraft type

change in Patrol Squadron Forty-Seven, the manual strategy did not assign them a ready-alert.

B. COST PERFORMANCE ANALYSIS

Seven more test runs were conducted to examine the effect of the pruning variable in relation to the cost of arriving at a solution. The first four tests demonstrated that increasing the pruning variable had little effect on scheduling for only one squadron (Table 5.5). This is because there is no competition for time or inspection teams resources. In this test the changing K had no effect on the answer.

Three tests were conducted with two squadrons that had overlapping at-home periods and were competing for the same inspection teams. Table 5.6 summarizes results. Notice that processing time is significantly increased though the number of events being scheduled only doubled from six to twelve. Early in the implementation some doubt arose as to whether slow computation was due to programming errors or whether the size of the pruning variable was wrong. I initially used values of the latter

Table 5.5 SQUADRON COST COMPARISON (Schedule/Performance)

vp9 Period 1	OPTIMAL COMPLETION	K = 0	K = .5	K = 1.0	K = 1.5
Event Name	Date	Difference	Difference	Difference	Difference
Pre-NTPI	20Feb86	+17	+17	+17	+17
NTPI	21Mar86	+7	+7	+7	+7
Ready-Alert 1st	30Apr86	0	0	0	0
NATOPS	7Feb86	0	0	0	0
Ready-Alert 2nd	30Jun86	0	0	0	0
CTPI pre-MRCI	11Jul86	-1	-1	-1	-1
MRCI	10Aug86	+4	+4	+4	+4
CI	10Sep86	+9	+9	+9	+9
Process Time (cpu sec.)		152.92	205.48	230.80	239.03
Successors Generated		59	247	247	247

Table 5.6 TEAM COST COMPARISON (Schedule/Performance)

Period 1	OPTIMAL COMPLETION	K = 0	K = 0	K = 0
vp40				
Event Name	Date	Difference	Difference	Difference
Pre-NTPI	13Jan86	-10	-7	-10
NTPI	4Feb86	-5	-5	-11
Ready-Alert 1st	31Dec85	0	0	0
NATOPS	8Nov85	+7	+7	0
Ready-Alert 2nd	28Feb86	0	0	0
CTPI pre-MRCI	11Apr86	-7	0	-7
MRCI	1May86	-8	+7	0
CI	27Sun86	-14	0	0
vp9				
Event Name	Date	Difference	Difference	Difference
Pre-NTPI	14Feb86	-10	-7	-7
NTPI	5Mar86	-5	-6	-10
Ready-Alert 1st	31Mar86	0	0	0
NATOPS	25Apr86	-7	-7	-7
Ready-Alert 2nd	30Jun86	0	0	0
CTPI pre-MRCI	11Jul86	-1	0	0
MRCI	10Aug86	-3	-3	-3
CI	25Sep86	-13	-15	-13
Process Time (cpu seconds)		240.97	864.18	3869.0
Successors Generated		183	1083	3579

ranging from 100 to 1000; a version with 1000 ran for thirty-one days. I reduced the value to zero and I had an answer in less than thirty minutes. This pruned everything worse than the best state one the agenda and was effectively hill-climbing. Continued testing of the pruning variable showed that marked differences in performance occurred with increments as small as 0.5. When the pruning variable was adjusted to 2.0, the number of agenda items rapidly increased with each pick of the best state. At twelve hours the previously sufficient 500K heap was exhausted though only six inspections for **search2** had been tentatively scheduled.

VI. CONCLUSION AND RECOMMENDATIONS

The results of the program are encouraging. It seems feasible that a computerized scheduling system can assist in the development of the annual training plan for a Patrol Air Wing.

Translation of the Training Officer's expertise was eased by the declarative nature of Prolog. The procedural steps of the manual method were implemented straightforwardly.

Modularizing the control structures for searching permits other search algorithms to be substituted. However, time was limited and I wanted to maintain a good environment for comparison of results due to other factors.

Incorporating the maintenance schedule in scheduling could easily be done in the existing rules through heuristics. While every attempt was made to accurately translate the Training Officer's methodologies some improvement could be made in scheduling around holidays: delaying an inspection following a major holiday is preferable. Avoiding the Christmas holiday period is also preferable (i.e., 19 December - 4 January). Scheduling the Command Inspection on the last working day of the ORE vulnerability will require additional heuristics.

The program presently starts from a database file containing the initialization data. A user interface can be built in which events could be added to the database or modifications made as necessary. This would allow the system to build the remaining schedule around a preferred start time for an event.

Nopath search proved that it is not necessary to keep track of the path lists to arrive at a satisfactory solution. However, discovering how to set the pruning variable K came

with trial and error. The user should carefully consider adjustments made to the pruning variable, comparing the quality of the solution against the time willing to wait for that solution.

A more appealing output could be done for the program. Symbolic names could be changed to more familiar acronyms. Gant chart presentations could also be produced.

It is recommended that further interviews be conducted with the PATWING TEN Maintenance Officer as well as with LCDR Sanford and the PATWING TEN Training Officer. Once fully implemented, it is recommended the program be tested against an on-going scheduling problem to recheck accuracy of the solution and feasibility of its use at the work site. It is further recommended alternate search strategies be explored, such as hierarchical ones, which could reduce the time to find a solution.

If the program were satisfactorily enhanced it should be usable by them all [Ref. 10]. This would include providing partitioned heuristics particular to each Patrol Air Wing scheduling algorithm. This could encourage more efficient utilization of common resources.

APPENDIX A - SOURCE CODE

SCHEDULER MODULE

This is the driver for the program **scheduler**. To execute, load Prolog with increased parameters. Then load the file **scheduler**.

```
% prolog -h 500
C-Prolog 1.5+
| ?- [scheduler].
```

The **scheduler** intitiates itself and consults the files **calendar**, **database**, **generator**, **depthsearch**, and **utilities**. Upon completion of **search1**, the files **estimator**, **nopathsearch**, **cost**, **earmark**, and **schedule_writer** are consulted. Calls are made to **search2** and **search3** which complete the scheduling process. The answer is produced as a list using **schedule_writer**. Details of this driver and the rest of the code are in Chapter IV.

```
go:-
        read_to_file,
        starttime,
        database_conversion,
        processtime(database_conversion,_),
        generate_trialperiods,
        processtime(generating_trialperiods,_),
        search1(Answer1),
        bringin,
        goalstatetotal,
        processtime(computing_goalstate,_),
        search2(Answer2),
        search3(Answer3),
        final_count,
        union(Answer2,Answer3,Answer4),
        earmark_dropdeaddates(Answer4),
        union(Answer1,Answer4,Answer),
        schedule_writer(Answer),
        display_statistics,
        told,
        halt.
go:-
        write('UNABLE TO DETERMINE SCHEDULE.  VERFIY DATABASE INPUT AND
        TRY AGAIN.'),nl,!.

% search1 is top level predicate to make ready assignments
search1(ReadyL):-
        write('Scheduling the Ready Alerts...'),nl,
        depthsearch([],ReadyL),!,
        write('SEARCH 1 RESULTS:'), nl,
        give_count(search1),
        write('(Ready Alert Roster): '),nl,
        prettyprint(ReadyL),
        build_readyevents(ReadyL),!.

search2(Answer) :-
```

```
        write('Scheduling the middle third of the schedule....'),nl,
        nopathsearch1([],Answer),!,
        write('SEARCH 2 RESULTS:'),nl,
        give_count(search2),
        prettyprint(Answer),!.

search3(Answer) :-
        write('Almost there, scheduling the LAST third of the schedule....'),nl,
        nopathsearch2([],Answer),!,
        write('SEARCH 3 RESULTS:'),nl,
        give_count(search3),
        prettyprint(Answer),!.

% Places output into file. FISCAL-YEAR-1986
read_to_file:-
        tell(FISCAL-YEAR-1986).

starttime:-
        timer,
        write('THE SCHEDULER IS PROCESSING....'),nl,nl,!.

timer:-
        system("date"),!.

give_count(Process):-
        processtime(Process,Time),
        sub_successorcount(SSK),
        retract(sub_successorcount(SSK)),
        assert(sub_successorcount(0)),
        write('==> Successors generated during '),write(Process),write(': '),
        write(SSK),nl,
        Average is Time/SSK,
        write('==> Average process time for successors in '),
        write(Process),write(': '),write(Average),write(' cpu seconds.'),nl,!.

final_count:-
        total_successorcount(TSK),
        X is cputime,
        Overall is X / TSK,
        write('Total successors = '),write(TSK),
        write(' and Overall Average Processing Time = '),write(Overall),
        write(' cpu seconds.'),nl,!.

successor_counter:-
        sub_successorcount(SSK),
        total_successorcount(TSK),
        retract(sub_successorcount(SSK)),
        retract(total_successorcount(TSK)),
        SSK2 is SSK + 1,
        TSK2 is TSK + 1,
        assert(sub_successorcount(SSK2)),
        assert(total_successorcount(TSK2)),!.

sub_successorcount(0).
```

```
total_successorcount(0).

processtime(Process,FT):-
      X is cputime,
      other_process_time(OPT),
      retract(other_process_time(OPT)),
      FT is X - OPT,
      write('The process for '),write(Process),
      write(' completed in: '),write(FT),write(' cpu seconds.'),nl,
      stars,
      assert(other_process_time(X)),!.

other_process_time(0).

display_statistics:-
      nl,nl,
      stars,stars,nl,
      statistics,nl,
      stars.

:-reconsult('database').
:-reconsult('calendar').
:-reconsult('generator').
:-reconsult('depthsearch').
:-reconsult('utilities').

bringin:-
      reconsult('estimator'),
      reconsult('nopathsearch'),
      reconsult('cost'),
      reconsult('earmark'),
      reconsult('schedule_writer'),nl,nl.

%Initializes the
scheduler
when it is loaded into a Prolog interpreter.
 :-go.
```

DATABASE MODULE

This file supports the data needed to generate the schedule. The dates are converted to numbers for ease of manipulation.

```
database_conversion:-
    write('Converting the database....'),nl,
    now,fail.
database_conversion:-
    year_to_schedule,
    deployment_database(D),
    database_conversion2(D),
    prior_schedule_database(D2),
    database_conversion3(D2).

year_to_schedule:-
    yearbegindate(X),
    yearenddate(Y),
    datetodaynumber(X,X2),
    datetodaynumber(Y,Y2),
    assertz(yearbegin(X2)),
    assertz(yearend(Y2)).
% Converts to days and places the prerequisites into the database.
database_conversion2([[Squadron,Eventname,Start,Finish]|Dbase]):-
    datetodaynumber(Start,Startdays),
    datetodaynumber(Finish,Finishdays),
    assertz(prerequisite(Squadron,Eventname,Startdays,Finishdays)),
    database_conversion2(Dbase).
database_conversion2([]):-
    abolish(deployment_database,1),!.

% Converts to days those events scheduled since last deployment
database_conversion3([[Squadron,Eventname,Start,Finish]|Dbase]):-
    datetodaynumber(Start,Startdays),
    datetodaynumber(Finish,Finishdays),
    assertz(priorevent(Squadron,Eventname,Startdays,Finishdays)),
    database_conversion3(Dbase).
database_conversion3([]):-
    abolish(prior_schedule_database,1),!.

% THE FOLLOWING MUST BE IN DATABASE FOR INITIALIZATION:

% Fiscal year scheduled.
yearbegindate([01,oct,1985]).
yearenddate([30,sep,1986]).

% PATWING TEN squadrons.
squadronlist([vp9,vp19,vp40,vp46,vp47,vp48,vp50]).

% List of events the schedule will handle
eventnames([tr1a,tr1b,ewp,ewn,en,ewc,ewm,ec]).

% The scheduled events which require an inspection team
teamevents([ewn,en,ewc,ewm,ec]).
```

```
% Blocks of time committed by long range planning.
deployment_database([
[vp9,ds0,[11,dec,1985],[10,jan,1986]],
[vp9,dv1,[01,jul,1986],[30,sep,1986]],
[vp9,dp1,[01,oct,1986],[15,nov,1986]],
[vp9,dr1,[16,nov,1986],[05,may,1987]],
[vp9,ds1,[06,may,1987],[05,apr,1987]],
%
[vp19,ds0,[11,mar,1985],[10,apr,1985]],
[vp19,dv1,[15,aug,1985],[15,nov,1985]],
[vp19,dp1,[16,nov,1985],[09,feb,1986]],
[vp19,dr1,[10,feb,1986],[10,aug,1986]],
[vp19,ds1,[11,aug,1986],[10,sep,1986]],
%
[vp40,ds0,[11,sep,1985],[10,oct,1985]],
[vp40,dv1,[01,apr,1986],[30,jun,1986]],
[vp40,dp1,[01,jul,1986],[09,aug,1986]],
[vp40,dr1,[10,aug,1986],[10,feb,1987]],
[vp40,ds1,[11,feb,1987],[10,mar,1987]],
%
[vp46,ds0,[11,aug,1985],[10,sep,1985]],
[vp46,dv1,[01,mar,1986],[31,may,1986]],
[vp46,dp1,[01,jun,1986],[09,jul,1986]],
[vp46,dr1,[10,jul,1986],[10,jan,1987]],
[vp46,ds1,[11,jan,1987],[10,feb,1987]],
%
[vp47,ds0,[11,feb,1985],[10,mar,1985]],
[vp47,dv1,[01,sep,1985],[30,nov,1985]],
[vp47,dp1,[01,dec,1985],[09,jan,1986]],
[vp47,dr1,[10,jan,1986],[10,jul,1986]],
[vp47,ds1,[11,jul,1986],[10,aug,1986]],
%
[vp48,ds0,[11,jan,1985],[10,feb,1985]],
[vp48,dv1,[01,aug,1985],[31,oct,1985]],
[vp48,dp1,[01,nov,1985],[09,dec,1985]],
[vp48,dr1,[10,dec,1985],[10,jun,1986]],
[vp48,ds1,[11,jun,1986],[10,jul,1986]],
%
[vp50,ds0,[11,jan,1986],[10,feb,1986]],
[vp50,dv1,[01,aug,1986],[31,oct,1986]],
[vp50,dp1,[01,nov,1986],[09,dec,1986]],
[vp50,dr1,[10,dec,1986],[10,jun,1987]],
[vp50,ds1,[11,jun,1987],[10,jul,1987]]
                        ]).


% The following would be completed by interface with the user or by using the
%   prior year's output. Only those events since the last safety standdown
%   of those squadron's currently at home would be entered into this part
%   of the database.
prior_schedule_database([
[vp19,tr1a,[01,sep,1985],[15,oct,1985]],      %last scheduled ready for year
[vp19,ewp,[11,apr,1985],[13,apr,1985]],
[vp19,ewn,[11,may,1985],[12,may,1985]],
```

```
[vp19,en,[23,jul,1985],[07,aug,1985]],
[vp19,ewc,[15,aug,1985],[19,aug,1985]],
[vp19,ewm,[19,sep,1985],[23,sep,1985]],
%
[vp46,ewp,[11,sep,1985],[13,sep,1985]],
%
[vp47,ewp,[11,mar,1985],[13,mar,1985]],
[vp47,ewn,[11,apr,1985],[12,apr,1985]],
[vp47,en,[27,jun,1985],[29,jun,1985]],
[vp47,ewc,[01,sep,1985],[05,sep,1985]],
%
[vp48,ewp,[11,feb,1985],[13,feb,1985]],
[vp48,ewn,[11,mar,1985],[12,mar,1985]],
[vp48,en,[11,may,1985],[24,may,1985]],
[vp48,ewc,[01,aug,1985],[05,aug,1985]],
[vp48,ewm,[01,sep,1985],[05,sep,1985]]
                    ]).
% Output from fiscal year 1985 schedule.  Flags the latest date an event
%    can be scheduled.
earmark(vp47,ewm,[15,oct,1985]).
earmark(vp46,ewn,[15,oct,1985]).
earmark(vp48,ec,[30,oct,1985]).
earmark(vp19,ec,[01,dec,1985]).
earmark(vp47,ec,[30,nov,1985]).
earmark(vp40,ewn,[30,nov,1985]).
earmark(vp46,en,[15,oct,1985]).
earmark(vp40,en,[30,jan,1986]).
earmark(vp9,ewn,[30,mar,1986]).
earmark(vp46,ewc,[15,mar,1986]).
earmark(vp50,ewn,[30,mar,1986]).
earmark(vp46,ewm,[15,apr,1986]).
earmark(vp40,ewc,[15,apr,1986]).
earmark(vp9,en,[30,apr,1986]).
earmark(vp46,ec,[15,jun,1986]).
earmark(vp40,ewm,[30,may,1986]).
earmark(vp50,en,[30,jun,1986]).
earmark(vp40,ec,[30,jul,1986]).
earmark(vp9,ewc,[15,jul,1986]).
earmark(vp48,en,[30,jul,1986]).
earmark(vp9,ewm,[15,aug,1986]).
earmark(vp47,en,[30,aug,1986]).
earmark(vp50,ewc,[15,sep,1986]).
earmark(vp9,ec,[30,sep,1986]).
earmark(vp19,en,[30,sep,1986]).
earmark(vp47,ewn,[30,sep,1986]).
earmark(vp50,ewm,[15,oct,1986]).
earmark(vp19,ewn,[02,nov,1986]).
earmark(vp50,ec,[15,nov,1986]).
earmark(vp48,ewn,[02,nov,1987]).
earmark(vp48,ewm,[30,jan,1987]).
earmark(vp48,ewc,[15,jan,1987]).
earmark(vp47,ewc,[15,apr,1987]).
earmark(vp19,ewm,[02,mar,1987]).
earmark(vp19,ewc,[15,feb,1987]).
```

GENERATOR MODULE

Computes all dates that qualify for each event to be scheduled. These are then put into the database as **trialperiods**.

```
% Driver for this program.
generate_trialperiods:-
    write('Generating possible inspection periods....'),nl,fail.
generate_trialperiods:-
    eventnames(EN),
    member(X,EN),
    generate_readies(X),
    generate_weapevals(X),
    generate_natops(X),
   generate_ore_weapevals(X),
    generate_ore_command(X),!.

% The ready alert periods are generated.
generate_readies(Eventname):-
    Eventname = tr1a,
    yearbegin(B),
    calc_start(B,0,S),
    daynumber_to_date(S,[Day,Month,Year]),
    Day = 1,
    daysofmonth(Month,D),
    calc_finish(S,D,F),
    assertz(trialperiod(tr1,S,F)),
    counter1(K),
    fail.
generate_readies(Eventname):-
    trialperiod(tr1,_,_),!.

% Pre-NTPI and NTPI trialperiods
generate_weapevals(Eventname):-
    (Eventname = ewp;
    Eventname = ewn),
    yearbegin(B),
    next_safedate(2,B,0,S),
    calc_finish(S,2,F),
    safeperiod(S,F),
    assertz(trialperiod(Eventname,S,F)),
    counter1(K),
    fail.
generate_weapevals(Eventname):-
    trialperiod(ewp,_,_),!,
    trialperiod(ewn,_,_),!.

% NATOPS periods are generated.
generate_natops(Eventname):-
    Eventname = en,
    yearbegin(B),
    next_safemonday(B,0,S),
    calc_finish(S,12,F),
    safeperiod_withweekends(S,F),
    assertz(trialperiod(en,S,F)),
```

```
        counter1(K),
        fail.
generate_natops(Eventname):-
        trialperiod(en,_,_),!.

% CTPI/pre-MRCI and MRCI periods.
generate_ore_weapevals(Eventname):-
        (Eventname = ewc;
        Eventname = ewm),
        yearbegin(B),
        next_safedate(4,B,0,S),
        calc_finish(S,4,F),
        safeperiod(S,F),
        assertz(trialperiod(Eventname,S,F)),
        counter1(K),
        fail.
generate_ore_weapevals(Eventname):-
        trialperiod(ewc,_,_),!,
        trialperiod(ewm,_,_),!.

% CI periods are generated.
generate_ore_command(Eventname):-
        yearbegin(B),
        next_safemonday(B,0,S),
        calc_finish(S,5,F),
        safeperiod(S,F),
        assertz(trialperiod(ec,S,F)),
        counter1(K),
        fail.
% Here a final count of the number of trialperiods generated is produced.
generate_ore_command(Eventname):-
        trialperiod(ec,_,_),!,
        count(N),
        write(N),
        retract(count(N)),
        assert(count(0)),
        write(' Trial Periods Generated.'),nl,!.

% Based on duration of the inspection, outputs the next date
%    discarding holidays and weekends.
next_safedate(Duration,DateIn,Delay,DateOut):-
        Duration = 2,
        date_calc(DateIn,Delay,DateOut),
        day_of_week(DateOut,Day),
        not(Day = friday),
        yearend(End),
        DateOut < End,
        safedate(DateOut).
next_safedate(Duration,DateIn,Delay,DateOut):-
        Duration = 4,
        date_calc(DateIn,Delay,DateOut),
        day_of_week(DateOut,Day),
        (Day = monday;
        Day = tuesday),
```

```
        yearend(End),
        DateOut < End,
        safedate(DateOut).
next_safedate(Duration,DateIn,Delay,DateOut):-
        Delay2 is Delay + 1,
        P is DateIn + Delay2,
        yearend(End),
        P < End,
        next_safedate(Duration,DateIn,Delay2,DateOut).

% Computes a non-holiday monday to start an inspection
next_safemonday(DateIn,Delay,DateOut):-
        date_calc(DateIn,Delay,DateOut),
        yearend(End),
        DateOut < End,
        safemonday(DateOut).
next_safemonday(DateIn,Delay,DateOut):-
        Delay2 is Delay + 1,
        date_calc(DateIn,Delay2,ProspectiveDate),
        yearend(End),
        ProspectiveDate < End,
        next_safemonday(DateIn,Delay2,DateOut).

%verifies monday is safe
safemonday(Date):-
        day_of_week(Date,Day),
        Day = monday,
        not(holiday(Date,Day,Holiday_name)).

% Verifies inspection period is not interrupted by a weekend or holiday.
safeperiod(Start,Finish):-
        Start = Finish,!.
safeperiod(Start,Finish):-
        D is Start + 1,
        safedate(D),
        safeperiod(D,Finish).

% Permits exclusion of weekends with holidays since impractical to schedule
%    a two week event over a holiday period anyway.
safeperiod_withweekends(Start,Finish):-
        Start = Finish.
safeperiod_withweekends(Start,Finish):-
        D is Start + 1,
        day_of_week(D,Day),
        not(holiday(D,Day,Holiday_name)),
        safeperiod_withweekends(D,Finish).

safedate(Date):-
                day_of_week(Date,Day),
                not(weekend(Day)),
                not(holiday(Date,Day,Holiday_name)),!.

% Calculates start of event regardless of weekends or holidays.
calc_start(Basedate,Delay,Start):-
```

```
        date_calc(Basedate,Delay,Start),
        yearend(End),
        Start < End + 31.
calc_start(Basedate,Delay,Start):-
        Delay2 is Delay + 1,
        P is Basedate + Delay2,
        yearend(End),
        P < End + 31,      %provides ready in case overlap with nextyear
        calc_start(Basedate,Delay2,Start).

%calculates finish of event
calc_finish(Start,Duration,Finish):-
        Days is Duration - 1,
        date_calc(Start,Days,Finish).
```

DEPTH FIRST COMPLEMENT MODULE

This complements the depthsearch program found in Appendix B. The ready alerts are tasked through the successor1 rules.

```
goalreached1([event(Sq,E,S,F)|L]):-
    (E = tr1a; E = tr1b),
    yearend(YE),
    F >= YE,
    length([event(Sq,E,S,F)|L],X),
    assertz(totalreadies(X)).          %to be used in goalreached2

% Successor1 predicates are used to find the next ready alert assignment.
%   One of the next three successors assigns the first ready of the year.
successor1(L,[event(Sq,tr1a,S,F)|L]):-
    L = [],
    last_ready_of_previous_year(Sq2,S),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    preferred_assignment(Sq,S,F).
successor1(L,[event(Sq,tr1a,S,F)|L]):-
    L = [],
    last_ready_of_previous_year(Sq2,S),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    alternate_assignment1(Sq,S,F).
successor1(L,[event(Sq,tr1a,S,F)|L]):-
    L = [],
    last_ready_of_previous_year(Sq2,S),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    alternate_assignment2(Sq,S,F).
successor1(L,[event(Sq,tr1a,S,F)|L]):-
    readySF(L,Sq2,S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1a,_,_),L)),
    preferred_assignment(Sq,S,F).
successor1(L,[event(Sq,tr1a,S,F)|L]):-
    readySF(L,Sq2,S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1a,_,_),L)),
    alternate_assignment3(Sq,S,F).
successor1(L,[event(Sq,tr1a,S,F)|L]):-
    readySF(L,Sq2,S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1a,_,_),L)),
    alternate_assignment1(Sq,S,F).
successor1(L,[event(Sq,tr1a,S,F)|L]):-
    readySF(L,Sq2,S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1a,_,_),L)),
    alternate_assignment4(Sq,S,F).
successor1(L,[event(Sq,tr1a,S,F)|L]):-
    readySF(L,Sq2,S,F),
    pick_nextsquadron(Sq2,Sq),
```

```
    not(member(event(Sq,tr1a,_,_),L)),
    alternate_assignment2(Sq,S,F).
successor1(L,[event(Sq,tr1b,S,F)|L]):-
    readySF(L,Sq2,S,F),
    pick_nextsquadron(Sq2,Sq),
    member(event(Sq,tr1a,S2,F2),L),
    not(member(event(Sq,tr1b,_,_),L)),
    S > F2 + 28,
    preferred_assignment(Sq,S,F).
successor1(L,[event(Sq,tr1b,S,F)|L]):-
    readySF(L,Sq2,S,F),
    pick_nextsquadron(Sq2,Sq),
    member(event(Sq,tr1a,S2,F2),L),
    not(member(event(Sq,tr1b,_,_),L)),
    S > F2 + 28,
    alternate_assignment2(Sq,S,F).
successor1([event(Sq2,E,S1,F1)|L],
      [event(Sq,tr1a,S,F),event(Sq2,E,S1,F2)|L]):-
    split_start(S1,F1,S,F2),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1a,_,_),L)),
    preferred_assignment(Sq,S,F).
successor1([event(Sq2,E,S1,F1)|L],
      [event(Sq,tr1a,S,F),event(Sq2,E,S1,F2)|L]):-
    split_start(S1,F1,S,F2),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1a,_,_),L)),
    alternate_assignment3(Sq,S,F).
successor1([event(Sq2,E,S1,F1)|L],
      [event(Sq,tr1a,S,F),event(Sq2,E,S1,F2)|L]):-
    split_start(S1,F1,S,F2),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1a,_,_),L)),
    alternate_assignment1(Sq,S,F).
successor1([event(Sq2,E,S1,F1)|L],
      [event(Sq,tr1a,S,F),event(Sq2,E,S1,F2)|L]):-
    split_start(S1,F1,S,F2),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1a,_,_),L)),
    alternate_assignment4(Sq,S,F).
successor1([event(Sq2,E,S1,F1)|L],
      [event(Sq,tr1a,S,F),event(Sq2,E,S1,F2)|L]):-
    split_start(S1,F1,S,F2),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1a,_,_),L)),
    alternate_assignment2(Sq,S,F).
% The next two rules assign a squadron's second ready alert if needed.
successor1([event(Sq2,E,S1,F1)|L],
      [event(Sq,tr1b,S,F),event(Sq2,E,S1,F2)|L]):-
```

```
    split_start(S1,F1,S,F2),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1b,_,_),L)),
    preferred_assignment(Sq,S,F).
successor1([event(Sq2,E,S1,F1)|L],
      [event(Sq,tr1b,S,F),event(Sq2,E,S1,F2)|L]):-
    split_start(S1,F1,S,F2),
    readyF(S,F),
    pick_nextsquadron(Sq2,Sq),
    not(member(event(Sq,tr1b,_,_),L)),
    alternate_assignment2(Sq,S,F).

% successor1 subordinate rules.
pick_nextsquadron(LastReadySquadron,NextReadySquadron):-
    squadronlist(SL),
    member(NextReadySquadron,SL),
    not(LastReadySquadron = NextReadySquadron).

% Used by first three successor1 rules.
last_ready_of_previous_year(LastReadySquadron,NewStart):-
    xbagof(RF1,readyfinishes1(tr1a,RF1),List1),
    xbagof(RF2,readyfinishes1(tr1b,RF2),List2),
    union(List1,List2,List3),
    max(List3,MF),
    (priorevent(LastReadySquadron,tr1a,_,MF);
    priorevent(LastReadySquadron,tr1b,_,MF)),
    NewStart is MF + 1,!.

readyfinishes1(ReadyName,ReadyFinish):-
    priorevent(Sq,ReadyName,ReadyStart,ReadyFinish).

readySF(ListofReadies,LastReadySquadron,NewStart,NewFinish):-
    last_ready(ListofReadies,LastReadySquadron,NewStart),
    readyF(NewStart,NewFinish).

last_ready([event(LastReadySquadron,ReadyName,S,F)|ListofReadies],
      LastReadySquadron,NewStart):-
      NewStart is F + 1,!.

% NewStart is first of month.
readyF(NewStart,NewFinish):-
    trialperiod(tr1,NewStart,NewFinish),!.
% NewStart is middle of month
readyF(NewStart,NewFinish):-
    trialperiod(tr1,S,NewFinish),
    S > NewStart,
    NewFinish < NewStart + 61,!.

% Ready is no earlier than third full month after deployment.
%   Ready is not during ORE vumerability period.
preferred_assignment(Sq,S,F):-
    prerequisite(Sq,ds0,S2,F2),
    S >= F2 + 59,
```

```
        prerequisite(Sq,dv1,S3,F3),
        F < S3,!.

% Ready is at least one month after post deployment safety standdown,
%    but before ORE vumerability period.
alternate_assignment1(Sq,S,F):-
        prerequisite(Sq,ds0,S2,F2),
        S >= F2 + 30,
        prerequisite(Sq,dv1,S3,F3),
        F < S3,!.

% Ready is no earlier than third full month after deployment.
%    Ready can be during first month of ORE vumerability period.
alternate_assignment2(Sq,S,F):-
        prerequisite(Sq,ds0,S2,F2),
        S >= F2 + 59,
        prerequisite(Sq,dv1,S3,F3),
        F < F3 - 59,!.

% Ready is no earlier than third full month after deployment.
alternate_assignment3(Sq,S,F):-
        prerequisite(Sq,ds1,S2,F2),
        S >= F2 + 59.

% Ready is at least one month after post deployment safety standdown,
alternate_assignment4(Sq,S,F):-
        prerequisite(Sq,ds1,S2,F2),
        S >= F2 + 30.

% Divides three months between two ready alerts.
split_start(OldStart,OldFinish,SplitStart,SplitFinish):-
        X is OldFinish - OldStart,
        X =< 31,
        SplitFinish is OldFinish + 15,
        SplitStart is SplitFinish + 1,!.
```

ESTIMATOR MODULE

Output used in **goalreached2** and **goalreached3** of **search2** and **search3** respectively. Estimates number of events that will be scheduled between the safety standdown and beginning of the ore vumerability period of each squadron as well as the number of events for the ore vumerability period.

```
goalstatetotal:-
    asserta(eventsubtotal1(0)),
    asserta(eventsubtotal2(0)),
    goalstatetotal2.
goalstatetotal:-
    eventsubtotal1(ST1),
    eventsubtotal2(ST2),
    write('Search 2 has '),write(ST1),write(' events to schedule.'),nl,
    write('Search 3 has '),write(ST2),write(' events to schedule.'),nl,nl,!.

% Totals optimum number of events to be scheduled during the fiscal year.
goalstatetotal2:-
    squadronlist(SL),
    member(Sq,SL),
    maxevents1(Sq),fail.
goalstatetotal2:-
    squadronlist(SL),
    member(Sq,SL),
    maxevents2(Sq),fail.

% When period 1 within fiscal year
maxevents1(Sq):-
    prerequisite(Sq,ds0,S,F),
    yearbegin(YB),
    F >= YB,
    prerequisite(Sq,dv1,S2,F2),
    yearend(YE),
    S2 =< YE,
    eventsubtotal1(ST),
    SUM is ST + 3,
    retract(eventsubtotal1(ST)),
    asserta(eventsubtotal1(SUM)),!.
% When period 1 start less than year start
maxevents1(Sq):-
    prerequisite(Sq,ds0,S,F),
    yearbegin(YB),
    F < YB,
    prerequisite(Sq,dv1,S2,F2),
    yearend(YE),
    YB < S2,
    S2 =< YE,
    count_priorevents(Sq,N),
    N2 is 3 - N,
    eventsubtotal1(ST),
    SUM is ST + N2,
    retract(eventsubtotal1(ST)),
    asserta(eventsubtotal1(SUM)),!.
% When period 1 finish greater than yearend
```

```
maxevents1(Sq):-
    prerequisite(Sq,ds1,S,F),
    yearbegin(YB),
    yearend(YE),
    YB < F, F < YE,
    readyevent(Sq,E,S2,F2),
    S2 > F,
    Range is YE - F,
    estimate2(Range),!.
maxevents1(Sq):-
    prerequisite(Sq,ds1,S,F),
    yearbegin(YB),
    yearend(YE),
    YB < F, F < YE,
    not(readyevent(Sq,E,S2,F2)),
    Range is YE - F,
    estimate3(Range),!.

% When period 2 within the fiscal year
maxevents2(Sq):-
    prerequisite(Sq,dv1,S,F),
    yearbegin(YB), S >= YB,
    yearend(YE),
    F =< YE, eventsubtotal2(ST),
    SUM is ST + 3,
    retract(eventsubtotal2(ST)),
    asserta(eventsubtotal2(SUM)),!.
maxevents2(Sq):-
    prerequisite(Sq,dv1,S,F),
    yearbegin(YB),
    S < YB, YB < F,
    count_priorevents(Sq,N),
    N2 is 6 - N,
    eventsubtotal2(ST),
    SUM is ST + N2,
    retract(eventsubtotal2(ST)),
    asserta(eventsubtotal2(SUM)),!.
maxevents2(Sq):-
    prerequisite(Sq,dv1,S,F),
    yearend(YE),
    S < YE, YE < F,
    Range is YE - S,
    estimate4(Range),!.

% Counts the events that were scheduled for a period in the previous year.
count_priorevents(Sq,N):-
    priorevent(Sq,E,S,F),
    checkevent(E),
    counter3, fail.
count_priorevents(Sq,N):-
    count3(N),
    retract(count3(N)), asserta(count3(0)),!.

checkevent(E):-
```

```
    not(E = tr1a), not(E = tr1b),!.

%estimates the number of events when the period is open at the year's end.
estimate1(Range):-
    Range =< 31,
    eventsubtotal1(ST),
    SUM is ST + 1,
    retract(eventsubtotal1(ST)),
    asserta(eventsubtotal1(SUM)),!.
estimate1(Range):-
    31 =< Range,
    Range =< 61,
    eventsubtotal1(ST),
    SUM is ST + 2,
    retract(eventsubtotal1(ST)),
    asserta(eventsubtotal1(SUM)),!.
estimate1(Range):-
    Range > 61,
    eventsubtotal1(ST),
    SUM is ST + 3,
    retract(eventsubtotal1(ST)),
    asserta(eventsubtotal1(SUM)),!.

estimate2(Range):-
    Range =< 82,
    eventsubtotal1(ST),
    SUM is ST + 1,
    retract(eventsubtotal1(ST)),
    asserta(eventsubtotal1(SUM)),!.
estimate2(Range):-
    Range > 82,
    eventsubtotal1(ST),
    SUM is ST + 3,
    retract(eventsubtotal1(ST)),
    asserta(eventsubtotal1(SUM)),!.

estimate3(Range):-
    Range =< 20,
    eventsubtotal1(ST),
    SUM is ST + 1,
    retract(eventsubtotal1(ST)),
    asserta(eventsubtotal1(SUM)),!.
estimate3(Range):-
    Range > 20,
    eventsubtotal1(ST),
    SUM is ST + 3,
    retract(eventsubtotal1(ST)),
    asserta(eventsubtotal1(SUM)),!.

estimate4(Range):-
    eventsubtotal2(ST),
    SUM is ST + 2,
    retract(eventsubtotal2(ST)),
    asserta(eventsubtotal2(SUM)),!.
```

```
% Allows user to selectively modify output without having to modify database
modify_squadronlist:-
    retract(squadronlist(SL)),
    assert(squadronlist([vp40,vp9,vp50,vp47,vp46,vp48,vp19])).
```

NOPATHSEARCH COMPLEMENT MODULE

Successor rules for **search2** and **search3**. These are called by the **nopathsearch** code to produced the transition states from which the goal path is developed. The **successor2** rules are for finding pre-NTPI, NTPI, and NATOPS successors. The CTPI/pre-MRCI, MRCI, and CI are solved through **successor3** rules.

```
% Scheds pre-NTPI before ready alert when no earlier than second full month.
successor2(L,[event(Sq,ewp,S,F)|L]):-
    squadronlist(List), member(Sq,List),
    not(priorevent(Sq,ewp,_,_)),
    not(member(event(Sq,ewp,_,_),L)),
    prerequisite(Sq,ds0,S1,F1),
    readyevent(Sq,tr1a,RS,RF),
    extractsquadronSF(Sq,L,SFlist),
    earmark(Sq,ewn,DDD),
    DDD2 is DDD - 23,
    trialperiod(ewp,S,F),
    F =< DDD2,
    S >= F1 + 21, F =< RS - 22,
    WU is RS - 1,                    %sets upper boundary of window
    window_protection(S,WU,SFlist),
    team_serial(ewp,S,F,L).
% Scheds pre-NTPI after ready if second ready is 30 days before ORE
successor2(L,[event(Sq,ewp,S,F)|L]):-
    squadronlist(List), member(Sq,List),
    not(priorevent(Sq,ewp,_,_)),
    not(member(event(Sq,ewp,_,_),L)),
    prerequisite(Sq,ds0,S1,F1),
    readyevent(Sq,tr1a,RS,RF),
    prerequisite(Sq,dv1,S2,F2),
    readyevent(Sq,tr1b,RbS,RbF),
    RbS > S2 - 32,                   %readyb in last month before dv1
    extractsquadronSF(Sq,L,SFlist),
    earmark(Sq,ewn,DDD),
    DDD2 is DDD - 23,
    trialperiod(ewp,S,F), F =< DDD2,
    S > RF, F < S2 - 60,             %gives 7 day window for ewp
    WU is S2 - 32,                   %upper boundary is day before readyb
    window_protection(S,WU,SFlist),
    team_serial(ewp,S,F,L).
% Schedules pre-NTPI after first ready and before second ready when
%   second ready is 45 days before ORE vurnerability period.
successor2(L,[event(Sq,ewp,S,F)|L]):-
    squadronlist(List), member(Sq,List),
    not(priorevent(Sq,ewp,_,_)),
    not(member(event(Sq,ewp,_,_),L)),
    prerequisite(Sq,ds0,S1,F1),
    readyevent(Sq,tr1a,RS,RF),
    prerequisite(Sq,dv1,S2,F2),
    readyevent(Sq,tr1b,RbS,RbF),
    RbS < S2 - 32, RbS > S2 - 48,
    extractsquadronSF(Sq,L,SFlist),
    earmark(Sq,ewn,DDD),
    DDD2 is DDD - 23,
    trialperiod(ewp,S,F),
```

```
        F =< DDD2,
        S > RF,
        F < S2 - 70, WU is S2 - 48,     %upper boundary is day before readyb
        window_protection(S,WU,SFlist),
        team_serial(ewp,S,F,L).
% Schedules pre-NTPI after first ready and before second ready when
%    second ready is 60 days before ORE vurnerability period.
successor2(L,[event(Sq,ewp,S,F)|L]):-
        squadronlist(List),
        member(Sq,List),
        not(priorevent(Sq,ewp,_,_)),
        not(member(event(Sq,ewp,_,_),L)),
        prerequisite(Sq,ds0,S1,F1),
        readyevent(Sq,tr1a,RS,RF),
        prerequisite(Sq,dv1,S2,F2),
        readyevent(Sq,tr1b,RbS,RbF),
        RbS < S2 - 48,
        RbS > S2 - 62,
        extractsquadronSF(Sq,L,SFlist),
        earmark(Sq,ewn,DDD),
        DDD2 is DDD - 23,
        trialperiod(ewp,S,F),
        F =< DDD2,
        S > RF,
        F < S2 - 80,
        WU is S2 - 62,                  %upper boundary is day before readyb
        window_protection(S,WU,SFlist),
        team_serial(ewp,S,F,L).
% Schedules pre-NTPI after first ready when there is no second ready
successor2(L,[event(Sq,ewp,S,F)|L]):-
        squadronlist(List),
        member(Sq,List),
        not(priorevent(Sq,ewp,_,_)),
        not(member(event(Sq,ewp,_,_),L)),
        prerequisite(Sq,ds0,S1,F1),
        readyevent(Sq,tr1a,RS,RF),
        prerequisite(Sq,dv1,S2,F2),
        not(readyevent(Sq,tr1b,RbS,RbF)),
        extractsquadronSF(Sq,L,SFlist),
        earmark(Sq,ewn,DDD),
        DDD2 is DDD - 23,
        trialperiod(ewp,S,F),
        F =< DDD2,
        S > RF,
        F < S2 - 30,
        WU is S2 - 1,                   %upper boundary is day before dv1
        window_protection(S,WU,SFlist),
        team_serial(ewp,S,F,L).
% Schedules pre-NTPI when full deployment in fiscal year
successor2(L,[event(Sq,ewp,S,F)|L]):-
        squadronlist(List),
        member(Sq,List),
        prerequisite(Sq,ds1,S1,F1),
        not(another(Sq,ewp,F1,L)),
```

```
        extractsquadronSF(Sq,L,SFlist),
        earmark(Sq,ewn,DDD),
        DDD2 is DDD - 23,
        trialperiod(ewp,S,F),
        F =< DDD2,
        S > F1 + 21,
        yearend(YE),
        F < YE - 22,
        window_protection(S,YE,SFlist),
        team_serial(ewp,S,F,L).
% Schedules NTPI after pre-NTPI during period 1.
successor2(L,[event(Sq,ewn,S,F)|L]):-
        squadronlist(List), member(Sq,List),
        (priorevent(Sq,ewp,S1,F1);
        member(event(Sq,ewp,S1,F1),L)),
        not(priorevent(Sq,ewn,_,_)),
        not(member(event(Sq,ewn,_,_),L)),
        extractsquadronSF(Sq,L,SFlist),
        earmark(Sq,ewn,DDD),
        trialperiod(ewn,S,F),
        F =< DDD, S > F1 + 14,
        prerequisite(Sq,dv1,S2,F2),
        F < S2 - 31, WL is F1 + 1,
        window_protection(WL,F,SFlist),
        team_serial(ewn,S,F,L).
% Schedules NTPI during period 2.
successor2(L,[event(Sq,ewn,S,F)|L]):-
        not(L = []),
        squadronlist(List), member(Sq,List),
        prerequisite(Sq,ds1,S1,F1),
        member(event(Sq,ewp,S2,F2),L),
        S2 > F1,
        not(another(Sq,ewn,F1,L)),
        extractsquadronSF(Sq,L,SFlist),
        deleteitems([[S2,F2]],SFlist,SFlist2),
        earmark(Sq,ewn,DDD),
        trialperiod(ewn,S,F),
        F =< DDD, S > F2 + 14,
        yearend(YE), F =< YE,
        WL is F2 + 1,
        window_protection(WL,F,SFlist2),
        team_serial(ewn,S,F,L).
% Schedules NATOPS during period 1.
successor2(L,[event(Sq,en, S,F)|L]):-
        squadronlist(List), member(Sq,List),
        not(priorevent(Sq,en,_,_)),
        not(member(event(Sq,en,_,_),L)),
        prerequisite(Sq,ds0,S1,F1),
        extractsquadronSF(Sq,L,SFlist),
        earmark(Sq,en,DDD),
        trialperiod(en,S,F),
        F =< DDD, S > F1,
        prerequisite(Sq,dv1,S2,F2),
        F < S2,
```

```
    window_protection2(Sq,S,F,L),
    event_serial(S,F,SFlist),
    team_serial(en,S,F,L).
% Schedules NATOPS during period 2.
successor2(L,[event(Sq,en, S,F)|L]):-
    squadronlist(List),
    member(Sq,List),
    prerequisite(Sq,ds1,S1,F1),
    not(another(Sq,en,F1,L)),
    extractsquadronSF(Sq,L,SFlist),
    earmark(Sq,en,DDD),
    trialperiod(en,S,F),
    F =< DDD,
    S > F1,
    yearend(YE),
    F =< YE,
    window_protection2(Sq,S,F,L),
    event_serial(S,F,SFlist),
    team_serial(en,S,F,L).

% Schedules CI in ORE vumerability period.
successor3(L,[event(Sq,ec, S,F)|L]):-
    squadronlist(List),
    member(Sq,List),
    not(priorevent(Sq,ec,_,_)),
    not(member(event(Sq,ec,_,_),L)),
    prerequisite(Sq,dv1,S1,F1),
    extractsquadronSF(Sq,L,SFlist),
    earmark(Sq,ec,DDD),
    trialperiod(ec,S,F),
    F =< DDD,
    S > S1 + 60,
    F =< F1,
    event_serial(S,F,SFlist),
    team_serial(ec,S,F,L).
% Schedules CTPI when no ready alert encroaching ORE vumerability period.
successor3(L,[event(Sq,ewc,S,F)|L]):-
    squadronlist(List),
    member(Sq,List),
    not(priorevent(Sq,ewc,_,_)),
    not(member(event(Sq,ewc,_,_),L)),
    not(readypresent(Sq)),
    prerequisite(Sq,dv1,S1,F1),
    extractsquadronSF(Sq,L,SFlist),
    earmark(Sq,ewc,DDD),
    trialperiod(ewc,S,F),
    F =< DDD,
    S >= S1,
    F =< S1 + 10,
    event_serial(S,F,SFlist),
    team_serial(ewc,S,F,L).
% Schedules CTPI after a ready alert in ORE vumerability period.
successor3(L,[event(Sq,ewc,S,F)|L]):-
    squadronlist(List),
```

```
    member(Sq,List),
    not(priorevent(Sq,ewc,_,_)),
    not(member(event(Sq,ewc,_,_),L)),
    readypresent(Sq),
    prerequisite(Sq,dv1,S1,F1),
    extractsquadronSF(Sq,L,SFlist),
    earmark(Sq,ewc,DDD),
    trialperiod(ewc,S,F),
    F =< DDD,
    S >= S1 + 15,
    F =< S1 + 41,
    event_serial(S,F,SFlist),
    team_serial(ewc,S,F,L).
% Regardless if ready present or not schedules MRCI after CTPI.
successor3(L,[event(Sq,ewm,S,F)|L]):-
    squadronlist(List),
    member(Sq,List),
    not(priorevent(Sq,ewm,_,_)),
    not(member(event(Sq,ewm,_,_),L)),
    (priorevent(Sq,ewc,S1,F1);
    member(event(Sq,ewc,S1,F1),L)),
    extractsquadronSF(Sq,L,SFlist),
    earmark(Sq,ewm,DDD),
    trialperiod(ewm,S,F),
    F =< DDD,
    S >= F1 + 21,
    F =< F1 + 41,                    %gives 3 week window
    event_serial(S,F,SFlist),
    team_serial(ewm,S,F,L).
```

The following are subordinate rules used in **successor2** rules and **successor3** rules.

```
% Tests the relationships between periods of time.
interferes(EventStart,EventFinish,SFlist):-
    not(SFlist = []),
    member([OtherStart,OtherFinish],SFlist),
    ((OtherStart =< EventFinish,
    OtherStart >= EventStart);
    (OtherFinish =< EventFinish,
    OtherFinish >= EventStart)),!.
interferes(EventStart,EventFinish,SFlist):-
    not(SFlist = []),
    member([OtherStart,OtherFinish],SFlist),
    ((EventStart =< OtherFinish,
    EventStart >= OtherStart);
    (EventFinish =< OtherFinish,
    EventFinish >= OtherStart)),!.

% Provides period between two events that is free from interference by other events.
window_protection(Lower,Upper,SFlist):-
    not(interferes(Lower,Upper,SFlist)),!.

window_protection2(Sq,S,F,L):-
    priorevent(Sq,ewp,S1,F1),
```

```
        member(event(Sq,ewn,S2,F2),L),
        not(interferes(S,F,[F1,S2])),!.
window_protection2(Sq,S,F,L):-
        member(event(Sq,ewp,S1,F1),L),
        member(event(Sq,ewn,S2,F2),L),
        not(interferes(S,F,[F1,S2])),!.
window_protection2(Sq,S,F,L):-
        priorevent(Sq,ewp,S1,F1),
        S > F1 + 40,!.
window_protection2(Sq,S,F,L):-
        member(event(Sq,ewp,S1,F1),L),
        (F < S1;
        S > F1 + 40),!.
window_protection2(Sq,S,F,L):-
        not(priorevent(Sq,ewp,S1,F1)),
        not(member(event(Sq,ewp,S2,F2),L)),!.

% For determining if ready is in ORE vurnerability period
readypresent(Sq):-
        prerequisite(Sq,dv1,S1,F1),
        readyevent(Sq,E,S2,F2),
        F2 > S1,!.

% Determines if event already scheduled in period 2.
another(Sq,Event,F,L):-
        member(event(Sq,Event,S2,F2),L),
        S2 > F, !.

% Creates list of event starts and finishes from one squadron.
extractsquadronSF(Sq,L,SFList):-
        xbagof([S,F],extractSF1(Sq,L,[S,F]),SFList).

extractSF1(Sq,L,[S,F]):-
        member(event(Sq,E,S,F),L).
extractSF1(Sq,L,[S,F]):-
        readyevent(Sq,tr1a,S,F).
extractSF1(Sq,L,[S,F]):-
        readyevent(Sq,tr1b,S,F).

% Checks that events of same squadron are consecutive.
event_serial(Start,Finish,SFlist):-
        not(interferes(Start,Finish,SFlist)),!.

% Checks that events with same teams are consecutive irrespective of squadron
team_serial(Event,Start,Finish,Oldstate):-
        extracteventSF(Event,Oldstate,SFlist),!,
        not(interferes(Start,Finish,SFlist)),!.

% Creates list of starts and finishes associated with one inspection team.
extracteventSF(Event,L,SFList):-
        xbagof([S,F],extractSF4(Event,L,[S,F]),SFList),!.

extractSF4(Event,L,[S,F]):-
        member(event(Sq,Event,S,F),L).
```

```
% Determines degree of optimization by purging agenda of selected items.
prunable(State,D,BestState,Dbest):-
    D > Dbest + 1,!.
prunable(State,D,BestState,Dbest):-
    check_permutation(State,BestState),!.

check_permutation(State1,State2):-
    subset(State1,State2),
    subset(State2,State1),!.

% Used in search2
goalreached2(State):-
    length(State,L1),
    eventsubtotal1(ST),
    L1 >= ST.

% Used in search3
goalreached3(State):-
    length(State,L1),
    eventsubtotal2(ST),
    L1 >= ST.

% Used in search2
eval1(State,Eval):-
    eventsubtotal1(ST),
    length(State,N2),
    X is ST - N2,
    Eval is X * 1.82793,!.

% Used in search3
eval2(State,Eval):-
    eventsubtotal2(ST),
    length(State,N2),
    X is ST - N2,
    Eval is X * 1.82793,!.
```

COST MODULE

These rules are called by **add_state1** and **add_state2** after each successor is determined. The combined cost of events distributed internal to each squadron and with respect to each inspection team are computed.

```
cost([],0):-!.
cost(State,Cost):-
      costS(State,Scost),
      costT(State,Tcost),
      Cost is Scost + Tcost,!.

costS([],0):-!.
costS(State,Scost):-
      member(event(Sq,_,_,_),State),
      squadroncost(Sq,State,ReducedState,Scost2),
      costS(ReducedState,Scost3),
      Scost is Scost2 + Scost3.

% Computes the cost associated with each squadron
squadroncost(Sq,State,ReducedState,Scost):-
      xbagof(Occurance,groupsquadron(Sq,State,Occurance),List),
      sortevent(List,SortedList),
      first(SortedList,X),
      piece_cost(start,X,Scost2),
      cost2(SortedList,Scost3),
      Scost is Scost2 + Scost3,
      deleteitems(List,State,ReducedState).

groupsquadron(Sq,State,event(Sq,E,S,F)):-
      member(event(Sq,E,S,F),State).

costT([],0):-!.
costT(State,Tcost):-
      member(event(_,Eventname,_,_),State),
      teamcost(Eventname,State,ReducedState,Tcost2),
      costT(ReducedState,Tcost3),
      Tcost is Tcost2 + Tcost3.

% Computes the costs related to inspection team load.
teamcost(Eventname,State,ReducedState,0):-
      Eventname = ewp,                    %local to squadron
      xbagof(Occurance,groupteam(Eventname,State,Occurance),List),
      deleteitems(List,State,ReducedState),!.
teamcost(Eventname,State,ReducedState,Tcost):-
      teamevents(TE),                     %found in database
      member(Eventname,TE),
      xbagof(Occurance,groupteam(Eventname,State,Occurance),List),
      sortevent(List,SortedList),
      cost2(SortedList,Tcost),
      deleteitems(List,State,ReducedState),!.

groupteam(Eventname,State,event(Sq,Eventname,S,F)):-
      member(event(Sq,Eventname,S,F),State).
```

```
% Sums costs between events.
cost2([X,Y|L],Cost):-
    not(L = []),
     piece_cost(X,Y,Cost2),
     cost2([Y|L],Cost3),
     Cost is Cost2 + Cost3.
cost2([X,Y|L],Cost):-
    L = [],
     piece_cost(X,Y,Cost).
cost2([X],0).
cost2([],0).

% Computes cost between two events when both occur after year start.
piece_cost(event(Sq1,E1,S1,F1),event(Sq2,E2,S2,F2),Cost):-
     difference_between_dates(S2,F1,D),
     Cost is exp((24-D)/12) + exp((D-24)/30).
% Cost of first event after post deployment safety stand down.
piece_cost(start,event(Sq,Event,S,F),Cost):-
     (Event = ewp;
     Event = en),
     prerequisite(Sq,ds0,S1,F1),
     yearbegin(YB),
     F1 > YB,
     D is S - F1,
     Cost is exp((D-24)/30).
% Cost of first event in period 2.
piece_cost(start,event(Sq,Event,S,F),Cost):-
     (Event = ewp;
     Event = en),
     prerequisite(Sq,ds1,S1,F1),
     yearbegin(YB),
     F1 > YB,
     yearend(YE),
     F1 < YE,
     D is S - F1,
     Cost is exp((D-24)/30).
% Cost of first event when period 1 starts before year start.
piece_cost(start,event(Sq,Event,S,F),Cost):-
     ((Event = ewp;
     Event = ewn);
     Event = en),
     prerequisite(Sq,ds0,S1,F1),
     yearbegin(YB),
     F1 < YB,
     D is S - YB,
     Cost is exp((D-24)/30).
% Cost of first event in ORE vumerability period.
piece_cost(start,event(Sq,Event,S,F),Cost):-
     (Event = ewc;
     Event = ec),
     prerequisite(Sq,dv1,S1,F1),
     yearbegin(YB),
     S1 >= YB,
```

```
    D is S - S1,
    Cost is exp((D-24)/30).
% Cost of first event when ORE vurnerability period starts before year start.
piece_cost(start,event(Sq,Event,S,F),Cost):-
    ((Event = ewc;
    Event = ewm);
    Event = ec),
    prerequisite(Sq,dv1,S1,F1),
    yearbegin(YB),
    S1 < YB,
    D is S - YB,
    Cost is exp((D-24)/30).
```

EARMARK MODULE

When the second and thrid subsearches are complete, **earmark_dropdeaddates** computes the latest date by which each inspection must be reevaluated. A flag, **earmark**, is placed in the database for use by the next schedule.

```
earmark_dropdeaddates([event(Squadron,Event,Start,Finish)|L]):-
    Event = ewn,
    retract(earmark(Squadron,ewn,OldDropDeadDate)),
    DropDeadDate is Finish + 548,
    daynumber_to_date(DropDeadDate,NewDropDeadDate),
    asserta(earmark(Squadron,ewn,NewDropDeadDate)),
    earmark_dropdeaddates(L).
earmark_dropdeaddates([event(Squadron,Event,Start,Finish)|L]):-
    Event = en,
    retract(earmark(Squadron,en,OldDropDeadDate)),
    DropDeadDate is Finish + 457,
    daynumber_to_date(DropDeadDate,NewDropDeadDate),
    asserta(earmark(Squadron,en,NewDropDeadDate)),
    earmark_dropdeaddates(L).
% The rest are all 24 month deadlines.
earmark_dropdeaddates([event(Squadron,Event,Start,Finish)|L]):-
    retract(earmark(Squadron,Event,OldDropDeadDate)),
    DropDeadDate is Finish + 730,
    asserta(earmark(Squadron,Event,NewDropDeadDate)),
    daynumber_to_date(DropDeadDate,NewDropDeadDate),
    earmark_dropdeaddates(L).
earmark_dropdeaddates([event(Squadron,Event,Start,Finish)|L]):-
    Event = ewp,
    earmark_dropdeaddates(L).
earmark_dropdeaddates([]):-!.
```

CALENDAR MODULE

This file contains various calendar utilities. Both the ability to manipulate numerical dates and Gregorian dates are manifested in them. Predicate names appended with the number "2" are reserved for the latter.

```
% determines day of week
day_of_week(Date,Day):-
    X is Date mod 7,
    daymod(Day,X).

day_of_week2(Date,Day):-
    datetodaynumber(Date,DayNumber),
    X is DayNumber mod 7,
    daymod(Day,X).

%Computes the number of days difference between two dates
difference_between_dates(Date1,Date2,Difference):-
    X is Date1 - Date2,
    Difference is abs(X),!.

difference_between_dates2(Date1,Date2,Difference):-
    datetodaynumber(Date1,Daynumber1),
    datetodaynumber(Date2,Daynumber2),
    X is Daynumber1 - Daynumber2,
    Difference is abs(X),!.

%uses 1600 as base date for ease of Gregorian correction.
datetodaynumber([Monthday,Month,Year],Daynumber):-
    Diff is Year - 1600,
    N is (Diff*365) + (Diff//4) + (Diff//400) - (Diff//100) + 1,
   days_so_far([Monthday,Month,Year],Days),
    Daynumber is N + Days.

days_so_far([Monthday,Month,Year],Days):-
    leapyear(Year),
    (Month = jan ; Month = feb),
    daysuntilmonth(Month,Days1),!,
    Days is Days1 + Monthday - 1.
days_so_far([Monthday,Month,Year],Days):-
    daysuntilmonth(Month,Days1),!,
    Days is Days1 + Monthday.

leapyear(Year):-
    X is Year mod 400, X = 0,!.
leapyear(Year):-
    X is Year mod 100,
    not(X = 0),
    Y is Year mod 4, Y = 0,!.

% Computes date after adding a positive or negative number
date_calc(DateIn,Days,DateOut):-
    DateOut is Days + DateIn,!.
```

```
date_calc2(DateIn,Days,DateOut):-
    datetodaynumber(DateIn,Daynumber),
    Daycount is Days + Daynumber,
    daynumber_to_date(Daycount,DateOut),!.

% Receives a daynumber representing the nth day since 01 Jan 1600 and
%   returns a date.
daynumber_to_date(DayCount,[Monthday,W,Year]):-
    Year is 1600 + (DayCount//365),
    Diff is Year - 1600,
    N is (Diff*365) + (Diff//4) + (Diff//400) - (Diff//100) + 1,
    not(N >= DayCount),
    not(leapyear(Year)),
    Days is DayCount - N,
    daysuntilmonth(X,Y),
    Y >= Days,
    premonth(W,X),!,
    daysuntilmonth(W,Z),
    Monthday is Days - Z,!.
daynumber_to_date(DayCount,[Monthday,W,Year]):-
    Year is 1600 + (DayCount//365),
    Diff is Year - 1600,
    N is (Diff*365) + (Diff//4) + (Diff//400) - (Diff//100) + 1,
    not(N >= DayCount),
    leapyear(Year),
    Days is DayCount - N + 1,
    daysuntilmonth(X,Y),
    Y >= Days,
    premonth(W,X),!,
    daysuntilmonth(W,Z),
    Monthday is Days - Z,!.
daynumber_to_date(DayCount,[Monthday,W,Year]):-
    Year1 is 1600 + (DayCount//365),
    Diff is Year1 - 1600,
    N is (Diff*365) + (Diff//4) + (Diff//400) - (Diff//100) + 1,
    N >= DayCount,
    Year is Year1 - 1,
    Diff2 is Year - 1600,
    N2 is (Diff2*365) + (Diff2//4) + (Diff2//400) - (Diff2//100)+ 1,
    not(leapyear(Year)),
    Days is DayCount - N2,
    daysuntilmonth(X,Y),
    Y >= Days,
    premonth(W,X),!,
    daysuntilmonth(W,Z),
    Monthday is Days - Z,!.
daynumber_to_date(DayCount,[Monthday,W,Year]):-
    Year1 is 1600 + (DayCount//365),
    Diff is Year1 - 1600,
    N is (Diff*365) + (Diff//4) + (Diff//400) - (Diff//100) + 1,
    N >= DayCount,
    Year is Year1 - 1,
    Diff2 is Year - 1600,
```

```
    N2 is (Diff2*365) + (Diff2//4) + (Diff2//400) - (Diff2//100)+ 1,
    leapyear(Year),
    Days is DayCount - N2 + 1,
    daysuntilmonth(X,Y),
    Y >= Days,
    premonth(W,X),!,
    daysuntilmonth(W,Z),
    Monthday is Days - Z,!.

holiday(Daynumber,Day,Holiday_name):-
    daynumber_to_date(Daynumber,Date),
    holiday_day(Date,Day,Holiday_name),!.

holiday_day([Date_day,Month,Year],Day,Holiday_name):-
    not(holiday_free_month(Month)),
    Day = monday,
    which_monday(Date_day,Monday_position),
    monday_holiday(Monday_position,Month,Holiday_name),!.
holiday_day([Date_day,Month,Year],Day,memorial_day):-
    Month = may,
    Day = monday,
    Date_day > 24.                    % last Monday in May
holiday_day([Date_day,Month,Year],Day,thanksgiving):-
    Month = nov,
    Day = thursday,
    Date_day > 21.

holiday_date([Date_day,Month,Year],Holiday_name):-
    not(holiday_free_month(Month)),
    day_of_week2([Date_day,Month,Year],Day),
    Day = monday,
    which_monday(Date_day,Monday_position),
    monday_holiday(Monday_position,Month,Holiday_name),!.
holiday_date([Date_day,Month,Year],memorial_day):-
    Month = may,
    day_of_week2([Date_day,Month,Year],Day),
    Day = monday,
    Date_day > 24,!.                  % last Monday in May
holiday_date([Date_day,Month,Year],thanksgiving):-
    Month = nov,
    day_of_week2([Date_day,Month,Year],Day),
    Day = thursday,
    Date_day > 21,!.

holiday_day([1,jan,Year],Day,new_years).          % New Year
holiday_day([4,jul,Year],Day,independence_day).   % Independence Day
holiday_day([11,nov,Year],Day,veterans_day).      % Veterans Day
holiday_day([25,dec,Year],Day,christmas).         % Christmas
holiday_date([1,jan,Year],new_years).             % New Year
holiday_date([4,jul,Year],independence_day).      % Independence Day
holiday_date([11,nov,Year],veterans_day).         % Veterans Day
holiday_date([25,dec,Year],christmas).            % Christmas

which_monday(Date_day,Monday_position):-
```

```
        (Date_day <  8,  Monday_position is 1);
        (Date_day >  7,Date_day <  15,  Monday_position is 2);
        (Date_day > 14,Date_day < 22,  Monday_position is 3);
        (Date_day > 21,Date_day < 29,  Monday_position is 4);
        (Date_day > 28,Date_day =< 31,  Monday_position is 5).

% Check to see if day is part of weekend
weekend(saturday).
weekend(sunday).

monday_holiday(3,jan,martin_luther_king).
monday_holiday(3,feb,washingtons_bday).
monday_holiday(1,sep,labor_day).
monday_holiday(2,oct,columbus_day).

% Days of each month
daysofmonth(jan,31).
daysofmonth(feb,28).
daysofmonth(mar,31).
daysofmonth(apr,30).
daysofmonth(may,31).
daysofmonth(jun,30).
daysofmonth(jul,31).
daysofmonth(aug,31).
daysofmonth(sep,30).
daysofmonth(oct,31).
daysofmonth(nov,30).
daysofmonth(dec,31).

daymod(sunday,2).
daymod(monday,3).
daymod(tuesday,4).
daymod(wednesday,5).
daymod(thursday,6).
daymod(friday,0).
daymod(saturday,1).

premonth(jan,feb).
premonth(feb,mar).
premonth(mar,apr).
premonth(apr,may).
premonth(may,jun).
premonth(jun,jul).
premonth(jul,aug).
premonth(aug,sep).
premonth(sep,oct).
premonth(oct,nov).
premonth(nov,dec).
premonth(dec,jan).

daysuntilmonth(jan,0).
daysuntilmonth(feb,31).
daysuntilmonth(mar,59).
daysuntilmonth(apr,90).
```

```
daysuntilmonth(may,120).
daysuntilmonth(jun,151).
daysuntilmonth(jul,181).
daysuntilmonth(aug,212).
daysuntilmonth(sep,243).
daysuntilmonth(oct,273).
daysuntilmonth(nov,304).
daysuntilmonth(dec,334).
daysuntilmonth(jan,365).          %added if premonth is dec

holiday_free_month(mar).
holiday_free_month(apr).
holiday_free_month(jun).
holiday_free_month(aug).
```

SCHEDULE WRITER MODULE

Prepares a simple output of the schedule in the form of a list.

```
schedule_writer(Answer):-
        dates(Answer,Output),
        prettyprint(Output).

dates(StateList,DatesOut):-
        sortevent(StateList,SortedEndingState),
        convert_to_dates(SortedEndingState,DatesOut).

convert_to_dates([event(Sq,Event,StartDays,FinishDays)|L],
  [event(Sq,Event,Date1,Date2)|Output]):-
        daynumber_to_date(StartDays,Date1),
        daynumber_to_date(FinishDays,Date2),
        convert_to_dates(L,Output).
convert_to_dates([],[]):-!.

% Sorts events by start dates
sortevent([],[]).
sortevent([event(Sq,E,S,F)|L1],L2):-
        sortevent(L1,L3),
        insert_event(event(Sq,E,S,F),L3,L2).

insert_event(event(Sq,E,S,F),[],[event(Sq,E,S,F)]).
insert_event(event(Sq,E,S,F),[event(Sq2,E2,S2,F2)|L],
        [event(Sq,E,S,F),event(Sq2,E2,S2,F2)|L]):-
        S < S2.
insert_event(event(Sq,E,S,F),[event(Sq2,E2,S2,F2)|L],
        [event(Sq2,E2,S2,F2)|L2]):-
        S >= S2,
        insert_event(event(Sq,E,S,F),L,L2).
```

UTILITIES MODULE

These are primarily list manipulating utilities which are used throughout the prototype [Ref. 7]. Also included are some counter routines for tracking various data during scheduling runs.

```
first([X|L],X).

last([X],X).
last([X|L],Y) :- last(L,Y).

member(X,[X|L]).
member(X,[Y|L]) :- member(X,L).

max([X],X).
max([X|L],X) :- max(L,M), X>M.
max([X|L],M) :- max(L,M), X=<M.

min([X],X).
min([X|L],X) :- min(L,M), X<M.
min([X|L],M) :- min(L,M), X>=M.

delete(X,[],[]).
delete(X,[X|L],M) :- !, delete(X,L,M).
delete(X,[Y|L],[Y|M]) :- delete(X,L,M).

append([],L,L).
append([X|L],L2,[X|L3]) :- append(L,L2,L3).

reverse(L,R) :- reverse2(L,[],R).
reverse2([],L,L) :- !.
reverse2([X|L],R,S) :- reverse2(L,[X|R],S).

singlemember(X,[X|L]) :- !.
singlemember(X,[Y|L]) :- singlemember(X,L).

deleteone(X,[],[]).
deleteone(X,[X|L],L) :- !.
deleteone(X,[Y|L],[Y|M]) :- deleteone(X,L,M).

subset([],L).
subset([X|L],L2) :- singlemember(X,L2), subset(L,L2).

% Deletes a set of items from another list
deleteitems([],L,L).
deleteitems([X|L],L2,L3) :- delete(X,L2,L4), deleteitems(L,L4,L3).

% Prints out a list with one item per line; useful for lists of lists
%    which can overflow the terminal line.
prettyprint([]) :- nl,!.
prettyprint([X|L]) :- write(X), nl, prettyprint(L).

union([],L,L).
union([X|L1],L2,L3) :- singlemember(X,L2), !, union(L1,L2,L3).
union([X|L1],L2,[X|L3]) :- union(L1,L2,L3).
```

```
xbagof(X,P,L):-bagof(X,P,L),!.
xbagof(X,P,[]):-!.

%counter for count_priorevents
counter3:-
      count3(K0),
      retract(count3(K0)),
      K is K0 + 1,
      asserta(count3(K)),!.
count3(0).

%counter for the file generator.
counter1(K):-
      count(K0),
      K is K0 + 1,
      retract(count(K0)),
      assert(count(K)),!.
count(0).

%counter for the file nopathsearch.
counter2(K):-
      count2(K0),
      K is K0 + 1,
      retract(count(K0)),
      assert(count(K)).
count2(0).
```

APPENDIX B - DEPTHSEARCH

This is a modified version of Prof. Rowe's **depthsearch**. It additionally counts the number of successors generated and asserts **readyevents** for use in building the rest of the schedule. This depth-first strategy is implemented by a call to **search1**. The answer is unioned with **search2** and **search3** answers for the final output.

```
depthsearch(Start,Ans) :-
     depthsearch2(Start,[Start],[Ans|Statelist]).
depthsearch2(State,Statelist,Statelist) :-
     goalreached1(State), !.
depthsearch2(State,Statelist,Ans) :-
     successor1(State,Newstate),
    not(member(Newstate,Statelist)),
     successor_counter,
    depthsearch2(Newstate,[Newstate|Statelist],Ans).

build_readyroster([]):-!.
build_readyroster([event(Sq,E,S,F)|L]):-
     asserta(readyevent(Sq,E,S,F)),
     build_readyroster(L).
```

APPENDIX C - NOPATHSEARCH

This is the modified version of Prof. Rowe's **nopathsearch** The successors generated are counted and the pruning action is recorded with supplemental **pick_best_state** rules. Each **beststate** is displayed to track the work being done bye the scheduler. The search program is used to schedule periods 1 and 2. Pre-NTPI, NTPI, and NATOPS are scheduled by the call to **search2**. The ORE vulnerability period is scheduled by **search3** (.i.e, CTPI/pre-MRCI, MRCI, and CI).

```
nopathsearch1(Start,State) :-
      cleandatabase,
      add_state1(Start),
      repeatifagenda,
      pick_best_state(State),
     write('------------------------------------------------------------'),nl,
     add_successors1(State),
      agenda(State,C,D),
     retract(agenda(State,C,D)),
      measurework.

nopathsearch2(Start,State) :-
     cleandatabase,
      add_state2(Start),
     repeatifagenda,
      pick_best_state(State),
     write('------------------------------------------------------------'),nl,
     add_successors2(State),
      agenda(State,C,D),
      retract(agenda(State,C,D)),
      measurework.

pick_best_state(State) :-
      asserta(beststate(dummy,dummy)),
      agenda(S,C,D),
      beststate(S2,D2),
      special_less_than(D,D2),
      retract(beststate(S2,D2)),
      asserta(beststate(S,D)), fail.
pick_best_state(State) :-
      countup(agenda(SX,CX,DX),AC),
      write('Item count BEFORE agenda pruned: '),write(AC),nl,fail.
pick_best_state(State) :-
      beststate(S,D),
      agenda(S2,C2,D2),
      not(S=S2),
      prunable(S2,D2,S,D),
      retract(agenda(S2,C2,D2)), fail.
pick_best_state(State) :-
      countup(agenda(SX,CX,DX),AC),
      write('Item count AFTER agenda pruned: '),write(AC),nl,fail.
pick_best_state(State) :-
      beststate(S,D),
```

```
    agenda(S,C,D),
    write('BestState:  COST= '),write(C), write(' ,D= '),write(D),nl,
    prettyprint(S),fail.
pick_best_state(State) :-
    beststate(State,D), retract(beststate(State,D)),
    not(D=dummy), !.

%used in search2
add_successors1(State) :- goalreached2(State), !.
add_successors1(State) :- successor2(State,Newstate),
    add_state1(Newstate), fail.
add_successors1(State) :- retract(agenda(State,C,D)),
    asserta(usedstate(State,C)),fail.

%used in search3
add_successors2(State) :- goalreached3(State), !.
add_successors2(State) :- successor3(State,Newstate),
    add_state2(Newstate), fail.
add_successors2(State) :- retract(agenda(State,C,D)),
    asserta(usedstate(State,C)),fail.

add_state1(Newstate) :-
    cost(Newstate,Cnew), !,
    agenda_check(Newstate,Cnew), !,
    usedstate_check(Newstate,Cnew), !,
    eval1(Newstate,Enew),
    D is Enew + Cnew,
    asserta(agenda(Newstate,Cnew,D)), !.
add_state1(Newstate) :-
    not(cost(Newstate,Cnew)),
    write('Warning: your cost function failed on path list '),
    write(Newstate), nl, !.
add_state1(Newstate) :-
    not(eval1(Newstate,Enew)),
    write('Warning: your evaluation function failed on state '),
    write(Newstate), nl, !.

add_state2(Newstate) :-
    cost(Newstate,Cnew), !,
    agenda_check(Newstate,Cnew), !,
    usedstate_check(Newstate,Cnew), !,
    eval2(Newstate,Enew),
    D is Enew + Cnew,
    asserta(agenda(Newstate,Cnew,D)), !.
add_state2(Newstate) :-
    not(cost(Newstate,Cnew)),
    write('Warning: your cost function failed on path list '),
    write(Newstate), nl, !.
add_state2(Newstate) :-
    not(eval2(Newstate,Enew)),
    write('Warning: your evaluation function failed on state '),
    write(Newstate), nl, !.

agenda_check(S,C) :-
```

```
        agenda(S,C2,D2),
        C<C2,
        retract(agenda(S,C2,D2)), !.
agenda_check(S,C) :-
        agenda(S,C2,D2), !, fail.
agenda_check(S,C).

usedstate_check(S,C) :-
        usedstate(S,C2), C<C2,
        retract(usedstate(S,C2)),
        asserta(usedstate(S,C)), !.
usedstate_check(S,C):- usedstate(S,C2),!,fail.
usedstate_check(S,C).

repeatifagenda.
repeatifagenda :-
        agenda(X,Z,W),
        repeatifagenda.

special_less_than(X,dummy) :- !.
special_less_than(X,Y) :-
        X<Y.

cleandatabase :-
        checkabolish(agenda,3),
        checkabolish(usedstate,2),
        checkabolish(beststate,2),
        checkabolish(counter,1).

checkabolish(P,N) :- abolish(P,N), !.
checkabolish(P,N).

measurework :-
        countup(agenda(X,C,D),NA),
        countup(usedstate(S,C),NB),
        write(NA),write(' incompletely examined state(s) and '),
        write(NB),write(' examined state(s)'), nl,!.

countup(P,N) :-
        asserta(counter(0)),
        call(P),
        counter(K),
        retract(counter(K)),
        K2 is K+1,
        asserta(counter(K2)), fail.
countup(P,N) :-
        counter(N),
        retract(counter(N)), !.
```

APPENDIX D - PRUNING VARIABLE K = 0

This demonstration conducts pruning permutations of present state and any state on the agenda worse than Dbest (cost + evaluation). Each beststate is listed. With it are shown the associated Cost, D value, and the number of items on the agenda before and after the pruning. At the completion of each subsearch the solution, amount of processing time for that subsearch, and the average time for processing each successor is given. The final answer is converted to dates and a statistical processing summary is given. These demonstrations were conducted on the ISI workstation.

```
isiv8% prolog -h 500
C-Prolog version 1.5
| ?- [scheduler].
database reconsulted 11164 bytes 1.18333 sec.
calendar reconsulted 9052 bytes 1.11667 sec.
generator reconsulted 4360 bytes 0.51667 sec.
depthsearch reconsulted 316 bytes 0.0666666 sec.
utilities reconsulted 3492 bytes 0.400001 sec.
Thu May 19 22:14:31 PDT 1988
THE SCHEDULER IS PROCESSING....

Converting the database....
The process for database_conversion completed in: 4.4 cpu seconds.
************************************************************
Generating possible inspection periods....
660 trial periods generated.
The process for generating_trialperiods completed in: 117.567 cpu seconds.
************************************************************
Scheduling the Ready Alerts...
SEARCH 1 RESULTS:
The process for search1 completed in: 10.95 cpu seconds.
************************************************************
==> Successors generated during search1: 19
==> Average process time for successors in search1: 0.576316 cpu seconds.
(Ready Alert Roster):
  event(vp48,tr1a,141212,141257)
  event(vp50,tr1b,141166,141211)
  event(vp9,tr1b,141120,141165)
  event(vp50,tr1a,141075,141119)
  event(vp9,tr1a,141044,141074)
  event(vp40,tr1b,141016,141043)
  event(vp46,tr1b,140985,141015)
  event(vp40,tr1a,140954,140984)
  event(vp46,tr1a,140908,140953)

estimator reconsulted 4364 bytes 0.483383 sec.
nopathsearch reconsulted 5076 bytes 0.633377 sec.
cost reconsulted 3640 bytes 0.433365 sec.
earmark reconsulted 716 bytes 0.133347 sec.
schedule_writer reconsulted 3016 bytes 0.400055 sec.

Search 2 has 16 events to schedule.
Search 3 has 15 events to schedule.
The process for computing_goalstate completed in: 3.2666 cpu seconds.
************************************************************
```

Scheduling the middle third of the schedule....
Item count BEFORE agenda pruned: 1
Item count AFTER agenda pruned: 1
BestState: COST= 0 ,D= 29.2469

Item count BEFORE agenda pruned: 77
Item count AFTER agenda pruned: 1
BestState: COST= 0.449329 ,D= 27.8683
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 70
Item count AFTER agenda pruned: 1
BestState: COST= 0.913888 ,D= 26.5049
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 65
Item count AFTER agenda pruned: 2
BestState: COST= 1.81872 ,D= 25.5818
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 57
Item count AFTER agenda pruned: 1
BestState: COST= 2.72356 ,D= 24.6587
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 50
Item count AFTER agenda pruned: 1
BestState: COST= 3.72356 ,D= 23.8308
event(vp9,ewp,141018,141019)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 39
Item count AFTER agenda pruned: 1
BestState: COST= 6.25438 ,D= 24.5337
event(vp19,en,141242,141253)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

```
---------------------------------------------------------
Item count BEFORE agenda pruned: 38
Item count AFTER agenda pruned: 1
BestState:  COST= 9.35168 ,D= 25.803
  event(vp48,en,141179,141190)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 35
Item count AFTER agenda pruned: 1
BestState:  COST= 12.5457 ,D= 27.1691
  event(vp47,ewp,141231,141232)
  event(vp48,en,141179,141190)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 37
Item count AFTER agenda pruned: 1
BestState:  COST= 17.31 ,D= 30.1055
  event(vp40,ewn,140944,140945)
  event(vp47,ewp,141231,141232)
  event(vp48,en,141179,141190)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 31
Item count AFTER agenda pruned: 1
BestState:  COST= 28.1203 ,D= 39.0879
  event(vp9,en,141095,141106)
  event(vp40,ewn,140944,140945)
  event(vp47,ewp,141231,141232)
  event(vp48,en,141179,141190)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

Item count BEFORE agenda pruned: 27
Item count AFTER agenda pruned: 1
BestState: COST= 35.6533 ,D= 44.7929
event(vp50,en,141137,141148)
event(vp9,en,141095,141106)
event(vp40,ewn,140944,140945)
event(vp47,ewp,141231,141232)
event(vp48,en,141179,141190)
event(vp19,en,141242,141253)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 24
Item count AFTER agenda pruned: 1
BestState: COST= 44.942 ,D= 52.2537
event(vp9,ewn,141034,141035)
event(vp50,en,141137,141148)
event(vp9,en,141095,141106)
event(vp40,ewn,140944,140945)
event(vp47,ewp,141231,141232)
event(vp48,en,141179,141190)
event(vp19,en,141242,141253)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 18
Item count AFTER agenda pruned: 1
BestState: COST= 43.8952 ,D= 49.3789
event(vp50,ewn,141070,141071)
event(vp9,ewn,141034,141035)
event(vp50,en,141137,141148)
event(vp9,en,141095,141106)
event(vp40,ewn,140944,140945)
event(vp47,ewp,141231,141232)
event(vp48,en,141179,141190)
event(vp19,en,141242,141253)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 12
Item count AFTER agenda pruned: 1

```
BestState:  COST= 56.3979 ,D= 60.0537
  event(vp40,en,140990,141001)
  event(vp50,ewn,141070,141071)
  event(vp9,ewn,141034,141035)
  event(vp50,en,141137,141148)
  event(vp9,en,141095,141106)
  event(vp40,ewn,140944,140945)
  event(vp47,ewp,141231,141232)
  event(vp48,en,141179,141190)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 8
Item count AFTER agenda pruned: 1
BestState:  COST= 61.8763 ,D= 63.7043
  event(vp46,en,140955,140966)
  event(vp40,en,140990,141001)
  event(vp50,ewn,141070,141071)
  event(vp9,ewn,141034,141035)
  event(vp50,en,141137,141148)
  event(vp9,en,141095,141106)
  event(vp40,ewn,140944,140945)
  event(vp47,ewp,141231,141232)
  event(vp48,en,141179,141190)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 5
Item count AFTER agenda pruned: 1
BestState:  COST= 234.042 ,D= 234.042
  event(vp47,ewn,141249,141250)
  event(vp46,en,140955,140966)
  event(vp40,en,140990,141001)
  event(vp50,ewn,141070,141071)
  event(vp9,ewn,141034,141035)
  event(vp50,en,141137,141148)
  event(vp9,en,141095,141106)
  event(vp40,ewn,140944,140945)
  event(vp47,ewp,141231,141232)
  event(vp48,en,141179,141190)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
```

```
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)


************************************************************
0 incompletely examined state(s) and 16 examined state(s)
SEARCH 2 RESULTS:
The process for search2 completed in: 835.983 cpu seconds.
************************************************************
==> Successors generated during search2: 592
==> Average process time for successors in search2: 1.41213 cpu seconds.

Almost there, scheduling the LAST third of the schedule....
Item count BEFORE agenda pruned: 1
Item count AFTER agenda pruned: 1
BestState:  COST= 0 ,D= 27.4189

------------------------------------------------------------
Item count BEFORE agenda pruned: 29
Item count AFTER agenda pruned: 2
BestState:  COST= 0.449329 ,D= 26.0403
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 27
Item count AFTER agenda pruned: 1
BestState:  COST= 0.898658 ,D= 24.6617
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 28
Item count AFTER agenda pruned: 1
BestState:  COST= 1.44747 ,D= 23.3826
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 26
Item count AFTER agenda pruned: 1
BestState:  COST= 3.84319 ,D= 23.9504
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 28
Item count AFTER agenda pruned: 1
BestState:  COST= 7.2388 ,D= 25.5181
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
```

```
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 24
Item count AFTER agenda pruned: 1
BestState:  COST= 13.2647 ,D= 29.716
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 23
Item count AFTER agenda pruned: 1
BestState:  COST= 24.1287 ,D= 38.7521
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 25
Item count AFTER agenda pruned: 1
BestState:  COST= 28.2346 ,D= 41.0301
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 24
Item count AFTER agenda pruned: 1
BestState:  COST= 137.947 ,D= 148.915
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
```

```
Item count BEFORE agenda pruned: 21
Item count AFTER agenda pruned: 1
BestState:  COST= 143.213 ,D= 152.352
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 17
Item count AFTER agenda pruned: 1
BestState:  COST= 156.898 ,D= 164.209
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 14
Item count AFTER agenda pruned: 1
BestState:  COST= 326.882 ,D= 332.365
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 10
Item count AFTER agenda pruned: 1
BestState:  COST= 329.21 ,D= 332.866
  event(vp40,ewm,141109,141112)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
```

```
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 5
Item count AFTER agenda pruned: 1
BestState:  COST= 338.061 ,D= 339.889
  event(vp9,ewm,141200,141203)
  event(vp40,ewm,141109,141112)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 1
Item count AFTER agenda pruned: 1
BestState:  COST= 342.779 ,D= 342.779
  event(vp50,ewm,141256,141259)
  event(vp9,ewm,141200,141203)
  event(vp40,ewm,141109,141112)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp19,ec,140927,140931)
  event(vp46,ewc,141046,141049)
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

************************************************************
0 incompletely examined state(s) and 15 examined state(s)
SEARCH 3 RESULTS:
The process for search3 completed in: 424.433 cpu seconds.
```

**********< ***

==> Successors generated during search3: 301

==> Average process time for successors in search3: 1.41008 cpu seconds.

TRAINING SCHEDULE FOR FISCAL YEAR 1986

event(vp47,ewm,[1,oct,1985],[4,oct,1985])
event(vp46,ewn,[1,oct,1985],[2,oct,1985])
event(vp48,ec,[7,oct,1985],[11,oct,1985])
event(vp46,tr1a,[16,oct,1985],[30,nov,1985])
event(vp40,ewp,[31,oct,1985],[1,nov,1985])
event(vp19,ec,[4,nov,1985],[8,nov,1985])
event(vp47,ec,[18,nov,1985],[22,nov,1985])
event(vp40,ewn,[21,nov,1985],[22,nov,1985])
event(vp40,tr1a,[1,dec,1985],[31,dec,1985])
event(vp46,en,[2,dec,1985],[13,dec,1985])
event(vp46,tr1b,[1,jan,1986],[31,jan,1986])
event(vp40,en,[6,jan,1986],[17,jan,1986])
event(vp40,tr1b,[1,feb,1986],[28,feb,1986])
event(vp9,ewp,[3,feb,1986],[4,feb,1986])
event(vp9,ewn,[19,feb,1986],[20,feb,1986])
event(vp9,tr1a,[1,mar,1986],[31,mar,1986])
event(vp46,ewc,[3,mar,1986],[6,mar,1986])
event(vp50,ewp,[3,mar,1986],[4,mar,1986])
event(vp50,ewn,[27,mar,1986],[28,mar,1986])
event(vp46,ewm,[31,mar,1986],[3,apr,1986])
event(vp40,ewc,[1,apr,1986],[4,apr,1986])
event(vp50,tr1a,[1,apr,1986],[15,may,1986])
event(vp9,en,[21,apr,1986],[2,may,1986])
event(vp46,ec,[5,may,1986],[9,may,1986])
event(vp40,ewm,[5,may,1986],[8,may,1986])
event(vp9,tr1b,[16,may,1986],[30,jun,1986])
event(vp40,ec,[2,jun,1986],[6,jun,1986])
event(vp50,en,[2,jun,1986],[13,jun,1986])
event(vp50,tr1b,[1,jul,1986],[15,aug,1986])
event(vp9,ewc,[7,jul,1986],[10,jul,1986])
event(vp48,en,[14,jul,1986],[25,jul,1986])
event(vp9,ewm,[4,aug,1986],[7,aug,1986])
event(vp47,en,[11,aug,1986],[22,aug,1986])
event(vp48,tr1a,[16,aug,1986],[30,sep,1986])
event(vp50,ewc,[2,sep,1986],[5,sep,1986])
event(vp47,ewp,[4,sep,1986],[5,sep,1986])
event(vp9,ec,[8,sep,1986],[12,sep,1986])
event(vp19,en,[15,sep,1986],[26,sep,1986])
event(vp47,ewn,[22,sep,1986],[23,sep,1986])
event(vp50,ewm,[29,sep,1986],[2,oct,1986])

****
****
STATISTICS:

atom space: 128K (in use: 34132, max. used: 34132)
aux. stack: 8K (in use: 0, max. used: 636)
trail: 64K (in use: 1620, max. used: 2296)
heap: 500K (in use: 153312, max. used: 156272)
global stack: 256K (in use: 55552, max. used: 54284)
local stack: 128K (in use: 8572, max. used: 32124)
Runtime: 1403.15 sec.

****
****

APPENDIX E - PRUNING VARIABLE K = 0.5

This is the first demonstration that actually uses the pruning action of the *prunable* rule. The only difference in the code that drives this demonstration and Appendix D is the incrementing of the pruning variable to 0.5.

```
isiv8% prolog -h 500
C-Prolog version 1.5
| ?- [scheduler].
database reconsulted 11164 bytes 1.15 sec.
calendar reconsulted 9052 bytes 1.13333 sec.
generator reconsulted 4360 bytes 0.533337 sec.
depthsearch reconsulted 316 bytes 0.0666671 sec.
utilities reconsulted 3492 bytes 0.383335 sec.

Thu May 19 23:54:42 PDT 1988
THE SCHEDULER IS PROCESSING....

Converting the database....
The process for database_conversion completed in: 7.46666 cpu seconds.
************************************************************
Generating possible inspection periods....
660 trial periods generated.
The process for generating_trialperiods completed in: 117.533 cpu seconds.
************************************************************
Scheduling the Ready Alerts...
SEARCH 1 RESULTS:
The process for search1 completed in: 10.9166 cpu seconds.
************************************************************
==> Successors generated during search1: 19
==> Average process time for successors in search1: 0.574559 cpu seconds.
(Ready Alert Roster):
  event(vp48,tr1a,141212,141257)
  event(vp50,tr1b,141166,141211)
  event(vp9,tr1b,141120,141165)
  event(vp50,tr1a,141075,141119)
  event(vp9,tr1a,141044,141074)
  event(vp40,tr1b,141016,141043)
  event(vp46,tr1b,140985,141015)
  event(vp40,tr1a,140954,140984)
  event(vp46,tr1a,140908,140953)

estimator reconsulted 4364 bytes 0.500061 sec.
nopathsearch reconsulted 4892 bytes 0.65007 sec.
cost reconsulted 3640 bytes 0.450104 sec.
earmark reconsulted 740 bytes 0.11676 sec.
schedule_writer reconsulted 3016 bytes 0.416672 sec.

Search 2 has 16 events to schedule.
Search 3 has 15 events to schedule.

The process for computing_goalstate completed in: 3.3833 cpu seconds.
************************************************************
Scheduling the middle third of the schedule....
```

Item count BEFORE agenda pruned: 1
Item count AFTER agenda pruned: 1
BestState: COST= 0 ,D= 29.2469

Item count BEFORE agenda pruned: 77
Item count AFTER agenda pruned: 21
BestState: COST= 0.449329 ,D= 27.8683
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 90
Item count AFTER agenda pruned: 14
BestState: COST= 0.913888 ,D= 26.5049
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 78
Item count AFTER agenda pruned: 13
BestState: COST= 1.81872 ,D= 25.5818
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 68
Item count AFTER agenda pruned: 9
BestState: COST= 2.72356 ,D= 24.6587
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 58
Item count AFTER agenda pruned: 4
BestState: COST= 3.72356 ,D= 23.8308
event(vp9,ewp,141018,141019)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 42
Item count AFTER agenda pruned: 3
BestState: COST= 3.75745 ,D= 23.8647
event(vp9,ewp,141019,141020)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 40
Item count AFTER agenda pruned: 2
BestState:  COST= 3.7925 ,D= 23.8997
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 38
Item count AFTER agenda pruned: 1
BestState:  COST= 3.82873 ,D= 23.9359
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 37
Item count AFTER agenda pruned: 1
BestState:  COST= 6.35955 ,D= 24.6388
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 36
Item count AFTER agenda pruned: 3
BestState:  COST= 9.45685 ,D= 25.9082
  event(vp48,en,141179,141190)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 35
Item count AFTER agenda pruned: 2
BestState:  COST= 9.55353 ,D= 26.0049
  event(vp47,ewp,141231,141232)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

```
---------------------------------------------------------
Item count BEFORE agenda pruned: 39
Item count AFTER agenda pruned: 1
BestState:  COST= 9.74815 ,D= 26.1995
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 38
Item count AFTER agenda pruned: 1
BestState:  COST= 12.8455 ,D= 27.4689
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 35
Item count AFTER agenda pruned: 6
BestState:  COST= 17.6098 ,D= 30.4053
  event(vp40,ewn,140944,140945)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 34
Item count AFTER agenda pruned: 5
BestState:  COST= 17.6343 ,D= 30.4298
  event(vp40,ewn,140943,140944)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
```

Item count BEFORE agenda pruned: 33
Item count AFTER agenda pruned: 4
BestState: COST= 17.6458 ,D= 30.4413
event(vp40,ewn,140948,140949)
event(vp48,en,141179,141190)
event(vp47,ewp,141230,141231)
event(vp19,en,141242,141253)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 32
Item count AFTER agenda pruned: 3
BestState: COST= 17.6738 ,D= 30.4693
event(vp40,ewn,140942,140943)
event(vp48,en,141179,141190)
event(vp47,ewp,141230,141231)
event(vp19,en,141242,141253)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 31
Item count AFTER agenda pruned: 2
BestState: COST= 17.6853 ,D= 30.4808
event(vp40,ewn,140949,140950)
event(vp48,en,141179,141190)
event(vp47,ewp,141230,141231)
event(vp19,en,141242,141253)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 30
Item count AFTER agenda pruned: 1
BestState: COST= 17.729 ,D= 30.5245
event(vp40,ewn,140941,140942)
event(vp48,en,141179,141190)
event(vp47,ewp,141230,141231)
event(vp19,en,141242,141253)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 29
Item count AFTER agenda pruned: 3
BestState:  COST= 28.0045 ,D= 38.9721
  event(vp9,en,141095,141106)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 27
Item count AFTER agenda pruned: 2
BestState:  COST= 28.2905 ,D= 39.2581
  event(vp9,en,141102,141113)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 26
Item count AFTER agenda pruned: 1
BestState:  COST= 28.2905 ,D= 39.2581
  event(vp9,en,141088,141099)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 25
Item count AFTER agenda pruned: 1
BestState:  COST= 34.5704 ,D= 43.71
  event(vp50,en,141137,141148)
  event(vp9,en,141088,141099)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
```

```
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 22
Item count AFTER agenda pruned: 1
BestState:  COST= 45.1027 ,D= 52.4144
  event(vp40,en,140990,141001)
  event(vp50,en,141137,141148)
  event(vp9,en,141088,141099)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 18
Item count AFTER agenda pruned: 1
BestState:  COST= 50.5811 ,D= 56.0649
  event(vp46,en,140955,140966)
  event(vp40,en,140990,141001)
  event(vp50,en,141137,141148)
  event(vp9,en,141088,141099)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 15
Item count AFTER agenda pruned: 3
BestState:  COST= 62.8396 ,D= 66.4954
  event(vp9,ewn,141039,141040)
  event(vp46,en,140955,140966)
  event(vp40,en,140990,141001)
  event(vp50,en,141137,141148)
  event(vp9,en,141088,141099)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
```

```
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 13
Item count AFTER agenda pruned: 3
BestState:  COST= 61.7779 ,D= 63.6058
  event(vp50,ewn,141070,141071)
  event(vp9,ewn,141039,141040)
  event(vp46,en,140955,140966)
  event(vp40,en,140990,141001)
  event(vp50,en,141137,141148)
  event(vp9,en,141088,141099)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 7
Item count AFTER agenda pruned: 2
BestState:  COST= 61.9884 ,D= 63.8163
  event(vp50,ewn,141069,141070)
  event(vp9,ewn,141039,141040)
  event(vp46,en,140955,140966)
  event(vp40,en,140990,141001)
  event(vp50,en,141137,141148)
  event(vp9,en,141088,141099)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 6
Item count AFTER agenda pruned: 1
BestState:  COST= 62.2188 ,D= 64.0468
  event(vp50,ewn,141068,141069)
  event(vp9,ewn,141039,141040)
  event(vp46,en,140955,140966)
```

```
  event(vp40,en,140990,141001)
  event(vp50,en,141137,141148)
  event(vp9,en,141088,141099)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------ ---
Item count BEFORE agenda pruned: 5
Item count AFTER agenda pruned: 1
BestState:  COST= 245.958 ,D= 245.958
  event(vp47,ewn,141249,141250)
  event(vp50,ewn,141068,141069)
  event(vp9,ewn,141039,141040)
  event(vp46,en,140955,140966)
  event(vp40,en,140990,141001)
  event(vp50,en,141137,141148)
  event(vp9,en,141088,141099)
  event(vp40,ewn,140941,140942)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

************************************************************
0 incompletely examined state(s) and 30 examined state(s)
SEARCH 2 RESULTS:
The process for search2 completed in: 1653.15 cpu seconds.
************************************************************
==> Successors generated during search2: 979
==> Average process time for successors in search2: 1.68861 cpu seconds.

Almost there, scheduling the LAST third of the schedule....
Item count BEFORE agenda pruned: 1
Item count AFTER agenda pruned: 1
BestState:  COST= 0 ,D= 27.4189

------------------------------------------------------------
Item count BEFORE agenda pruned: 29
Item count AFTER agenda pruned: 13
BestState:  COST= 0.449329 ,D= 26.0403
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
```

Item count BEFORE agenda pruned: 38
Item count AFTER agenda pruned: 10
BestState: COST= 0.898658 ,D= 24.6617
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

Item count BEFORE agenda pruned: 37
Item count AFTER agenda pruned: 3
BestState: COST= 1.44747 ,D= 23.3826
event(vp48,ec,140899,140903)
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

Item count BEFORE agenda pruned: 28
Item count AFTER agenda pruned: 4
BestState: COST= 1.77383 ,D= 23.709
event(vp48,ec,140913,140917)
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

Item count BEFORE agenda pruned: 28
Item count AFTER agenda pruned: 3
BestState: COST= 1.77383 ,D= 23.709
event(vp19,ec,140913,140917)
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

Item count BEFORE agenda pruned: 26
Item count AFTER agenda pruned: 4
BestState: COST= 3.84319 ,D= 23.9504
event(vp46,ewc,141046,141049)
event(vp48,ec,140899,140903)
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

Item count BEFORE agenda pruned: 31
Item count AFTER agenda pruned: 3
BestState: COST= 3.89799 ,D= 24.0052
event(vp46,ewc,141047,141050)
event(vp48,ec,140899,140903)
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

Item count BEFORE agenda pruned: 31
Item count AFTER agenda pruned: 2
BestState: COST= 4.16956 ,D= 24.2768
event(vp46,ewc,141046,141049)
event(vp19,ec,140913,140917)

```
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 27
Item count AFTER agenda pruned: 1
BestState:  COST= 4.22435 ,D= 24.3316
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 27
Item count AFTER agenda pruned: 2
BestState:  COST= 8.41187 ,D= 26.6911
  event(vp47,ec,140941,140945)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 26
Item count AFTER agenda pruned: 1
BestState:  COST= 8.61152 ,D= 26.8908
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 26
Item count AFTER agenda pruned: 1
BestState:  COST= 12.799 ,D= 29.2504
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 24
Item count AFTER agenda pruned: 2
BestState:  COST= 23.663 ,D= 38.2864
  event(vp9,ewc,141172,141175)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)
```

Item count BEFORE agenda pruned: 27
Item count AFTER agenda pruned: 1
BestState: COST= 24.0309 ,D= 38.6544
event(vp9,ewc,141173,141176)
event(vp47,ec,140941,140945)
event(vp48,ec,140899,140903)
event(vp46,ewc,141047,141050)
event(vp19,ec,140913,140917)
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

Item count BEFORE agenda pruned: 27
Item count AFTER agenda pruned: 1
BestState: COST= 28.0549 ,D= 40.8504
event(vp50,ewc,141229,141232)
event(vp9,ewc,141173,141176)
event(vp47,ec,140941,140945)
event(vp48,ec,140899,140903)
event(vp46,ewc,141047,141050)
event(vp19,ec,140913,140917)
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

Item count BEFORE agenda pruned: 26
Item count AFTER agenda pruned: 1
BestState: COST= 137.663 ,D= 148.63
event(vp46,ec,141109,141113)
event(vp50,ewc,141229,141232)
event(vp9,ewc,141173,141176)
event(vp47,ec,140941,140945)
event(vp48,ec,140899,140903)
event(vp46,ewc,141047,141050)
event(vp19,ec,140913,140917)
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

Item count BEFORE agenda pruned: 23
Item count AFTER agenda pruned: 1
BestState: COST= 142.928 ,D= 152.068
event(vp40,ec,141137,141141)
event(vp46,ec,141109,141113)
event(vp50,ewc,141229,141232)
event(vp9,ewc,141173,141176)
event(vp47,ec,140941,140945)
event(vp48,ec,140899,140903)
event(vp46,ewc,141047,141050)
event(vp19,ec,140913,140917)
event(vp40,ewc,141075,141078)
event(vp47,ewm,140893,140896)

```
---------------------------------------------------------
Item count BEFORE agenda pruned: 19
Item count AFTER agenda pruned: 1
BestState:  COST= 156.509 ,D= 163.82
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------
Item count BEFORE agenda pruned: 16
Item count AFTER agenda pruned: 1
BestState:  COST= 326.644 ,D= 332.127
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------
Item count BEFORE agenda pruned: 11
Item count AFTER agenda pruned: 4
BestState:  COST= 328.972 ,D= 332.628
  event(vp40,ewm,141109,141112)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

---------------------------------------------------------
Item count BEFORE agenda pruned: 9
Item count AFTER agenda pruned: 3
```

```
BestState:  COST= 329.02 ,D= 332.675
  event(vp40,ewm,141110,141113)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 8
Item count AFTER agenda pruned: 2
BestState:  COST= 329.068 ,D= 332.724
  event(vp40,ewm,141103,141106)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 7
Item count AFTER agenda pruned: 1
BestState:  COST= 329.151 ,D= 332.807
  event(vp40,ewm,141102,141105)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 6
Item count AFTER agenda pruned: 2
```

```
BestState:  COST= 340.369 ,D= 342.197
  event(vp9,ewm,141200,141203)
  event(vp40,ewm,141102,141105)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 2
Item count AFTER agenda pruned: 1
BestState:  COST= 340.686 ,D= 342.514
  event(vp9,ewm,141201,141204)
  event(vp40,ewm,141102,141105)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 1
Item count AFTER agenda pruned: 1
BestState:  COST= 345.326 ,D= 345.326
  event(vp50,ewm,141256,141259)
  event(vp9,ewm,141201,141204)
  event(vp40,ewm,141102,141105)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141229,141232)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp48,ec,140899,140903)
  event(vp46,ewc,141047,141050)
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)
```

****
0 incompletely examined state(s) and 26 examined state(s)
SEARCH 3 RESULTS:
The process for search3 completed in: 725.117 cpu seconds.
****
==> Successors generated during search3: 517
==> Average process time for successors in search3: 1.40255 cpu seconds.

TRAINING SCHEDULE FOR FISCAL YEAR 1986

event(vp47,ewm,[1,oct,1985],[4,oct,1985])
event(vp46,ewn,[1,oct,1985],[2,oct,1985])
event(vp48,ec,[7,oct,1985],[11,oct,1985])
event(vp46,tr1a,[16,oct,1985],[30,nov,1985])
event(vp19,ec,[21,oct,1985],[25,oct,1985])
event(vp40,ewp,[31,oct,1985],[1,nov,1985])
event(vp47,ec,[18,nov,1985],[22,nov,1985])
event(vp40,ewn,[18,nov,1985],[19,nov,1985])
event(vp40,tr1a,[1,dec,1985],[31,dec,1985])
event(vp46,en,[2,dec,1985],[13,dec,1985])
event(vp46,tr1b,[1,jan,1986],[31,jan,1986])
event(vp40,en,[6,jan,1986],[17,jan,1986])
event(vp40,tr1b,[1,feb,1986],[28,feb,1986])
event(vp9,ewp,[6,feb,1986],[7,feb,1986])
event(vp9,ewn,[24,feb,1986],[25,feb,1986])
event(vp9,tr1a,[1,mar,1986],[31,mar,1986])
event(vp50,ewp,[3,mar,1986],[4,mar,1986])
event(vp46,ewc,[4,mar,1986],[7,mar,1986])
event(vp50,ewn,[25,mar,1986],[26,mar,1986])
event(vp46,ewm,[31,mar,1986],[3,apr,1986])
event(vp40,ewc,[1,apr,1986],[4,apr,1986])
event(vp50,tr1a,[1,apr,1986],[15,may,1986])
event(vp9,en,[14,apr,1986],[25,apr,1986])
event(vp40,ewm,[28,apr,1986],[1,may,1986])
event(vp46,ec,[5,may,1986],[9,may,1986])
event(vp9,tr1b,[16,may,1986],[30,jun,1986])
event(vp40,ec,[2,jun,1986],[6,jun,1986])
event(vp50,en,[2,jun,1986],[13,jun,1986])
event(vp50,tr1b,[1,jul,1986],[15,aug,1986])
event(vp9,ewc,[8,jul,1986],[11,jul,1986])
event(vp48,en,[14,jul,1986],[25,jul,1986])
event(vp9,ewm,[5,aug,1986],[8,aug,1986])
event(vp47,en,[11,aug,1986],[22,aug,1986])
event(vp48,tr1a,[16,aug,1986],[30,sep,1986])
event(vp50,ewc,[2,sep,1986],[5,sep,1986])
event(vp47,ewp,[3,sep,1986],[4,sep,1986])
event(vp9,ec,[8,sep,1986],[12,sep,1986])
event(vp19,en,[15,sep,1986],[26,sep,1986])
event(vp47,ewn,[22,sep,1986],[23,sep,1986])
event(vp50,ewn,[29,sep,1986],[2,oct,1986])

```
************************************************************
************************************************************

atom space: 128K (in use: 34272, max. used: 34272)
aux. stack: 8K (in use: 0, max. used: 636)
trail: 64K (in use: 1736, max. used: 2520)
heap: 500K (in use: 165128, max. used: 165984)
global stack: 256K (in use: 56304, max. used: 55036)
local stack: 128K (in use: 8792, max. used: 32376)
Runtime:  2521.13 sec.

************************************************************
Total successors = 1515
Overall Average Processing Time = 1.66411 cpu seconds.
************************************************************
************************************************************
[ Prolog execution halted ]
```

APPENDIX F - PRUNING VARIABLE K = 1.0

Optimization carried one level higher with leaving all states in the agenda whose D value is less than or equal to Dbest + 1.0. Due to the length of the demonstration only representative segments are shown: The initial part of each subsearch, its summary, and the final answer and statistics is given.

```
isiv8% prolog -h 500
C-Prolog version 1.5
| ?- [scheduler].
database reconsulted 11164 bytes 1.16667 sec.
calendar reconsulted 9052 bytes 1.13333 sec.
generator reconsulted 4360 bytes 0.566669 sec.
depthsearch reconsulted 316 bytes 0.0666666 sec.
utilities reconsulted 3492 bytes 0.416668 sec.

Fri May 20 11:41:22 PDT 1988
THE SCHEDULER IS PROCESSING....

Converting the database....
The process for database_conversion completed in: 7.45 cpu seconds.
************************************************************
Generating possible inspection periods....
660 trial periods generated.
The process for generating_trialperiods completed in: 117.883 cpu seconds.
************************************************************
Scheduling the Ready Alerts...
SEARCH 1 RESULTS:
The process for search1 completed in: 10.9666 cpu seconds.
************************************************************
==> Successors generated during search1: 19
==> Average process time for successors in search1: 0.57719 cpu seconds.
(Ready Alert Roster):
  event(vp48,tr1a,141212,141257)
  event(vp50,tr1b,141166,141211)
  event(vp9,tr1b,141120,141165)
  event(vp50,tr1a,141075,141119)
  event(vp9,tr1a,141044,141074)
  event(vp40,tr1b,141016,141043)
  event(vp46,tr1b,140985,141015)
  event(vp40,tr1a,140954,140984)
  event(vp46,tr1a,140908,140953)

estimator reconsulted 4364 bytes 0.500092 sec.
nopathsearch reconsulted 4892 bytes 0.600098 sec.
cost reconsulted 3640 bytes 0.450058 sec.
earmark reconsulted 740 bytes 0.11673 sec.
schedule_writer reconsulted 3016 bytes 0.383423 sec.


Search 2 has 16 events to schedule.
Search 3 has 15 events to schedule.

The process for computing_goalstate completed in: 3.30005 cpu seconds.
************************************************************
```

```
**********************************************************

Scheduling the middle third of the schedule....
Item count BEFORE agenda pruned: 1
Item count AFTER agenda pruned: 1
BestState:  COST= 0 ,D= 29.2469

----------------------------------------------------------
Item count BEFORE agenda pruned: 77
Item count AFTER agenda pruned: 39
BestState:  COST= 0.449329 ,D= 27.8683
  event(vp46,ewn,140893,140894)

----------------------------------------------------------
Item count BEFORE agenda pruned: 108
Item count AFTER agenda pruned: 32
BestState:  COST= 0.913888 ,D= 26.5049
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

----------------------------------------------------------
Item count BEFORE agenda pruned: 96
Item count AFTER agenda pruned: 15
BestState:  COST= 1.81872 ,D= 25.5818
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

----------------------------------------------------------
Item count BEFORE agenda pruned: 70
Item count AFTER agenda pruned: 12
BestState:  COST= 2.72356 ,D= 24.6587
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

                    .
                    .
                    .
          78 BestStates skipped...
                    .
                    .
                    .

BestState:  COST= 44.6755 ,D= 50.1593
  event(vp50,ewn,141067,141068)
  event(vp9,ewn,141039,141040)
  event(vp9,en,141088,141099)
  event(vp50,en,141137,141148)
  event(vp40,ewn,140948,140949)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp40,ewp,140930,140931)
```

```
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 13
Item count AFTER agenda pruned: 1
BestState:  COST= 54.8518 ,D= 58.5077
  event(vp40,en.140990,141001)
  event(vp50,ewn,141067,141068)
  event(vp9,ewn,141039,141040)
  event(vp9,en,141088,141099)
  event(vp50,en,141137,141148)
  event(vp40,ewn,140948,140949)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp40,ewp,140930,140931)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 8
Item count AFTER agenda pruned: 1
BestState:  COST= 60.3303 ,D= 62.1582
  event(vp46,en,140955,140966)
  event(vp40,en,140990,141001)
  event(vp50,ewn,141067,141068)
  event(vp9,ewn,141039,141040)
  event(vp9,en,141088,141099)
  event(vp50,en,141137,141148)
  event(vp40,ewn,140948,140949)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp40,ewp,140930,140931)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 5
Item count AFTER agenda pruned: 1
BestState:  COST= 250.214 ,D= 250.214
  event(vp47,ewn,141249,141250)
  event(vp46,en,140955,140966)
  event(vp40,en,140990,141001)
  event(vp50,ewn,141067,141068)
  event(vp9,ewn,141039,141040)
  event(vp9,en,141088,141099)
```

```
  event(vp50,en,141137,141148)
  event(vp40,ewn,140948,140949)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp19,en,141242,141253)
  event(vp40,ewp,140930,140931)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)


**************************************************************
0 incompletely examined state(s) and 87 examined state(s)
SEARCH 2 RESULTS:
The process for search2 completed in: 4912.98 cpu seconds.
**************************************************************
==> Successors generated during search2: 3047
==> Average process time for successors in search2: 1.6124 cpu seconds.

Almost there, scheduling the LAST third of the schedule....
Item count BEFORE agenda pruned: 1
Item count AFTER agenda pruned: 1
BestState:  COST= 0 ,D= 27.4189

------------------------------------------------------------
Item count BEFORE agenda pruned: 33
Item count AFTER agenda pruned: 21
BestState:  COST= 0.449329 ,D= 26.0403
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 50
Item count AFTER agenda pruned: 17
BestState:  COST= 0.898658 ,D= 24.6617
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 48
Item count AFTER agenda pruned: 5
BestState:  COST= 1.44747 ,D= 23.3826
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 34
Item count AFTER agenda pruned: 6
BestState:  COST= 1.77383 ,D= 23.709
  event(vp48,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)
```

```
Item count BEFORE agenda pruned: 34
Item count AFTER agenda pruned: 7
BestState:  COST= 1.77383 ,D= 23.709
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)
---------- ----------------------------------------

                                .
                                .
                                .
                    32 BestStates skipped...
                                .
                                .
                                .
                                .

Item count BEFORE agenda pruned: 10
Item count AFTER agenda pruned: 2
BestState:  COST= 326.623 ,D= 330.279
  event(vp40,ewm,141116,141119)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141144,141148)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141222,141225)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp46,ewc,141047,141050)
  event(vp48,ec,140899,140903)
  event(vp19,ec,140927,140931)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 9
Item count AFTER agenda pruned: 1
BestState:  COST= 326.717 ,D= 330.372
  event(vp40,ewm,141102,141105)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141144,141148)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141222,141225)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp46,ewc,141047,141050)
  event(vp48,ec,140899,140903)
  event(vp19,ec,140927,140931)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 8
Item count AFTER agenda pruned: 2
BestState:  COST= 337.935 ,D= 339.763
```

```
  event(vp9,ewm,141200,141203)
  event(vp40,ewm,141102,141105)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141144,141148)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141222,141225)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp46,ewc,141047,141050)
  event(vp48,ec,140899,140903)
  event(vp19,ec,140927,140931)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 4
Item count AFTER agenda pruned: 1
BestState:  COST= 338.252 ,D= 340.08
  event(vp9,ewm,141201,141204)
  event(vp40,ewm,141102,141105)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141144,141148)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141222,141225)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp46,ewc,141047,141050)
  event(vp48,ec,140899,140903)
  event(vp19,ec,140927,140931)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 3
Item count AFTER agenda pruned: 3
BestState:  COST= 342.439 ,D= 342.439
  event(vp50,ewm,141249,141252)
  event(vp9,ewm,141201,141204)
  event(vp40,ewm,141102,141105)
  event(vp46,ewm,141074,141077)
  event(vp9,ec,141235,141239)
  event(vp40,ec,141144,141148)
  event(vp46,ec,141109,141113)
  event(vp50,ewc,141222,141225)
  event(vp9,ewc,141173,141176)
  event(vp47,ec,140941,140945)
  event(vp46,ewc,141047,141050)
  event(vp48,ec,140899,140903)
  event(vp19,ec,140927,140931)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)
```

---

2 incompletely examined state(s) and 42 examined state(s)
SEARCH 3 RESULTS:
The process for search3 completed in: 1511.98 cpu seconds.
****
==> Successors generated during search3: 1040
==> Average process time for successors in search3: 1.45383 cpu seconds.

TRAINING SCHEDULE FOR FISCAL YEAR 1986

event(vp47,ewm,[1,oct,1985],[4,oct,1985])
event(vp46,ewn,[1,oct,1985],[2,oct,1985])
event(vp48,ec,[7,oct,1985],[11,oct,1985])
event(vp46,tr1a,[16,oct,1985],[30,nov,1985])
event(vp19,ec,[4,nov,1985],[8,nov,1985])
event(vp40,ewp,[7,nov,1985],[8,nov,1985])
event(vp47,ec,[18,nov,1985],[22,nov,1985])
event(vp40,ewn,[25,nov,1985],[26,nov,1985])
event(vp40,tr1a,[1,dec,1985],[31,dec,1985])
event(vp46,en,[2,dec,1985],[13,dec,1985])
event(vp46,tr1b,[1,jan,1986],[31,jan,1986])
event(vp40,en,[6,jan,1986],[17,jan,1986])
event(vp40,tr1b,[1,feb,1986],[28,feb,1986])
event(vp9,ewp,[5,feb,1986],[6,feb,1986])
event(vp9,ewn,[24,feb,1986],[25,feb,1986])
event(vp9,tr1a,[1,mar,1986],[31,mar,1986])
event(vp50,ewp,[3,mar,1986],[4,mar,1986])
event(vp46,ewc,[4,mar,1986],[7,mar,1986])
event(vp50,ewn,[24,mar,1986],[25,mar,1986])
event(vp46,ewm,[31,mar,1986],[3,apr,1986])
event(vp40,ewc,[1,apr,1986],[4,apr,1986])
event(vp50,tr1a,[1,apr,1986],[15,may,1986])
event(vp9,en,[14,apr,1986],[25,apr,1986])
event(vp40,ewm,[28,apr,1986],[1,may,1986])
event(vp46,ec,[5,may,1986],[9,may,1986])
event(vp9,tr1b,[16,may,1986],[30,jun,1986])
event(vp50,en,[2,jun,1986],[13,jun,1986])
event(vp40,ec,[9,jun,1986],[13,jun,1986])
event(vp50,tr1b,[1,jul,1986],[15,aug,1986])
event(vp9,ewc,[8,jul,1986],[11,jul,1986])
event(vp48,en,[14,jul,1986],[25,jul,1986])
event(vp9,ewm,[5,aug,1986],[8,aug,1986])
event(vp47,en,[11,aug,1986],[22,aug,1986])
event(vp48,tr1a,[16,aug,1986],[30,sep,1986])
event(vp50,ewc,[26,aug,1986],[29,aug,1986])
event(vp47,ewp,[3,sep,1986],[4,sep,1986])
event(vp9,ec,[8,sep,1986],[12,sep,1986])
event(vp19,en,[15,sep,1986],[26,sep,1986])
event(vp50,ewm,[22,sep,1986],[25,sep,1986])
event(vp47,ewn,[22,sep,1986],[23,sep,1986])

```
**********************************************************
**********************************************************

atom space: 128K (in use: 34272, max. used: 34272)
aux. stack: 8K (in use: 0, max. used: 636)
trail: 64K (in use: 2032, max. used: 3384)
heap: 500K (in use: 188192, max. used: 188984)
global stack: 256K (in use: 56572, max. used: 55304)
local stack: 128K (in use: 8792, max. used: 32208)
Runtime:  6568.08 sec.

**********************************************************
Total successors = 4106;
Overall Average Processing Time = 1.59963 cpu seconds.
**********************************************************
**********************************************************

[ Prolog execution halted ]
```

APPENDIX G - PRUNING VARIABLE K = 1.5

This demonstration shows *search1* and *search2* in their entirety and gives a summary of *search3*.

```
isiv8% prolog -h 500
C-Prolog version 1.5
| ?- [scheduler].
database reconsulted 11164 bytes 1.16667 sec.
calendar reconsulted 9052 bytes 1.11667 sec.
generator reconsulted 4360 bytes 0.56667 sec.
depthsearch reconsulted 316 bytes 0.0500011 sec.
utilities reconsulted 3492 bytes 0.400002 sec.

Mon May 23 10:12:25 PDT 1988
THE SCHEDULER IS PROCESSING....

Converting the database....
The process for database_conversion completed in: 7.55 cpu seconds.
************************************************************
Generating possible inspection periods....
660 trial periods generated.
The process for generating_trialperiods completed in: 118.567 cpu seconds.
************************************************************
Scheduling the Ready Alerts...
SEARCH 1 RESULTS:
The process for search1 completed in: 11.0833 cpu seconds.
************************************************************
==> Successors generated during search1: 19
==> Average process time for successors in search1: 0.583332 cpu seconds.
************************************************************

(Ready Alert Roster):
  event(vp48,tr1a,141212,141257)
  event(vp50,tr1b,141166,141211)
  event(vp9,tr1b,141120,141165)
  event(vp50,tr1a,141075,141119)
  event(vp9,tr1a,141044,141074)
  event(vp40,tr1b,141016,141043)
  event(vp46,tr1b,140985,141015)
  event(vp40,tr1a,140954,140984)
  event(vp46,tr1a,140908,140953)

estimator reconsulted 4364 bytes 0.533356 sec.
nopathsearch reconsulted 4892 bytes 0.58342 sec.
cost reconsulted 3640 bytes 0.450043 sec.
earmark reconsulted 740 bytes 0.133438 sec.
schedule_writer reconsulted 3016 bytes 0.40007 sec.


Search 2 has 16 events to schedule.
Search 3 has 15 events to schedule.

The process for computing_goalstate completed in: 3.34998 cpu seconds.
************************************************************
Scheduling the middle third of the schedule....
```

```
Item count BEFORE agenda pruned: 1
Item count AFTER agenda pruned: 1
BestState:  COST= 0 ,D= 29.2469

---------------------------------------------------------------
Item count BEFORE agenda pruned: 77
Item count AFTER agenda pruned: 40
BestState:  COST= 0.449329 ,D= 27.8683
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 109
Item count AFTER agenda pruned: 40
BestState:  COST= 0.913888 ,D= 26.5049
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 104
Item count AFTER agenda pruned: 36
BestState:  COST= 1.81872 ,D= 25.5818
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 91
Item count AFTER agenda pruned: 21
BestState:  COST= 2.72356 ,D= 24.6587
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 70
Item count AFTER agenda pruned: 12
BestState:  COST= 3.72356 ,D= 23.8308
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 50
Item count AFTER agenda pruned: 14
BestState:  COST= 3.75745 ,D= 23.8647
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140923,140924)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

Item count BEFORE agenda pruned: 51
Item count AFTER agenda pruned: 16
BestState: COST= 3.7925 ,D= 23.8997
event(vp9,ewp,141020,141021)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 52
Item count AFTER agenda pruned: 18
BestState: COST= 3.82873 ,D= 23.9359
event(vp9,ewp,141021,141022)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 54
Item count AFTER agenda pruned: 22
BestState: COST= 6.25438 ,D= 24.5337
event(vp19,en,141242,141253)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 59
Item count AFTER agenda pruned: 23
BestState: COST= 6.28827 ,D= 24.5676
event(vp19,en,141242,141253)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 59
Item count AFTER agenda pruned: 24
BestState: COST= 6.32331 ,D= 24.6026
event(vp19,en,141242,141253)
event(vp9,ewp,141020,141021)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 59
Item count AFTER agenda pruned: 25
BestState: COST= 6.35955 ,D= 24.6388
event(vp19,en,141242,141253)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140923,140924)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 60
Item count AFTER agenda pruned: 27
BestState: COST= 2.81872 ,D= 24.7539
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 71
Item count AFTER agenda pruned: 20
BestState: COST= 3.72356 ,D= 23.8308
event(vp40,ewp,140923,140924)
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 58
Item count AFTER agenda pruned: 23
BestState: COST= 3.85262 ,D= 23.9598
event(vp40,ewp,140927,140928)
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 60
Item count AFTER agenda pruned: 25
BestState: COST= 3.88766 ,D= 23.9949
event(vp40,ewp,140928,140929)
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 61
Item count AFTER agenda pruned: 27
BestState: COST= 3.92389 ,D= 24.0311
event(vp40,ewp,140929,140930)

event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 62
Item count AFTER agenda pruned: 29
BestState: COST= 3.96135 ,D= 24.0686
event(vp40,ewp,140930,140931)
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 64
Item count AFTER agenda pruned: 33
BestState: COST= 6.25438 ,D= 24.5337
event(vp19,en,141242,141253)
event(vp40,ewp,140923,140924)
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 70
Item count AFTER agenda pruned: 35
BestState: COST= 6.38343 ,D= 24.6627
event(vp19,en,141242,141253)
event(vp40,ewp,140927,140928)
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 71
Item count AFTER agenda pruned: 36
BestState: COST= 6.41848 ,D= 24.6978
event(vp19,en,141242,141253)
event(vp40,ewp,140928,140929)
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 71
Item count AFTER agenda pruned: 37
BestState: COST= 6.45471 ,D= 24.734
event(vp19,en,141242,141253)
event(vp40,ewp,140929,140930)

event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 71
Item count AFTER agenda pruned: 38
BestState: COST= 6.49217 ,D= 24.7715
event(vp19,en,141242,141253)
event(vp40,ewp,140930,140931)
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 72
Item count AFTER agenda pruned: 39
BestState: COST= 2.85262 ,D= 24.7878
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 87
Item count AFTER agenda pruned: 29
BestState: COST= 3.85262 ,D= 23.9598
event(vp9,ewp,141018,141019)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 66
Item count AFTER agenda pruned: 31
BestState: COST= 3.88651 ,D= 23.9937
event(vp9,ewp,141019,141020)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 67
Item count AFTER agenda pruned: 33
BestState: COST= 3.92156 ,D= 24.0288
event(vp9,ewp,141020,141021)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 68
Item count AFTER agenda pruned: 35
BestState: COST= 3.95779 ,D= 24.065
event(vp9,ewp,141021,141022)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 70
Item count AFTER agenda pruned: 39
BestState: COST= 6.38344 ,D= 24.6627
event(vp19,en,141242,141253)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 75
Item count AFTER agenda pruned: 40
BestState: COST= 6.41733 ,D= 24.6966
event(vp19,en,141242,141253)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 75
Item count AFTER agenda pruned: 41
BestState: COST= 6.45237 ,D= 24.7316
event(vp19,en,141242,141253)
event(vp9,ewp,141020,141021)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 75
Item count AFTER agenda pruned: 42
BestState: COST= 6.48861 ,D= 24.7679
event(vp19,en,141242,141253)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 76
Item count AFTER agenda pruned: 43
BestState:  COST= 2.85262 ,D= 24.7878
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 86
Item count AFTER agenda pruned: 32
BestState:  COST= 3.75746 ,D= 23.8647
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 69
Item count AFTER agenda pruned: 35
BestState:  COST= 3.88651 ,D= 23.9937
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 71
Item count AFTER agenda pruned: 37
BestState:  COST= 3.92156 ,D= 24.0288
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 72
Item count AFTER agenda pruned: 39
BestState:  COST= 3.95779 ,D= 24.065
  event(vp40,ewp,140929,140930)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 73
Item count AFTER agenda pruned: 41
BestState:  COST= 3.99525 ,D= 24.1025
  event(vp40,ewp,140930,140931)
```

```
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 75
Item count AFTER agenda pruned: 45
BestState:  COST= 6.28827 ,D= 24.5676
  event(vp19,en,141242,141253)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 81
Item count AFTER agenda pruned: 47
BestState:  COST= 6.41733 ,D= 24.6966
  event(vp19,en,141242,141253)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 82
Item count AFTER agenda pruned: 48
BestState:  COST= 6.45237 ,D= 24.7316
  event(vp19,en,141242,141253)
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 82
Item count AFTER agenda pruned: 49
BestState:  COST= 6.48861 ,D= 24.7679
  event(vp19,en,141242,141253)
  event(vp40,ewp,140929,140930)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 82
Item count AFTER agenda pruned: 50
BestState:  COST= 6.52607 ,D= 24.8053
  event(vp19,en,141242,141253)
```

```
  event(vp40,ewp,140930,140931)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 83
Item count AFTER agenda pruned: 51
BestState:  COST= 2.88766 ,D= 24.8228
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 98
Item count AFTER agenda pruned: 41
BestState:  COST= 3.88766 ,D= 23.9949
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 77
Item count AFTER agenda pruned: 43
BestState:  COST= 3.92156 ,D= 24.0288
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 78
Item count AFTER agenda pruned: 45
BestState:  COST= 3.9566 ,D= 24.0638
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 79
Item count AFTER agenda pruned: 47
BestState:  COST= 3.99283 ,D= 24.1001
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 81
Item count AFTER agenda pruned: 51
BestState:  COST= 6.41848 ,D= 24.6978
  event(vp19,en,141242,141253)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 86
Item count AFTER agenda pruned: 52
BestState:  COST= 6.45237 ,D= 24.7316
  event(vp19,en,141242,141253)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 86
Item count AFTER agenda pruned: 53
BestState:  COST= 6.48742 ,D= 24.7667
  event(vp19,en,141242,141253)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 86
Item count AFTER agenda pruned: 54
BestState:  COST= 6.52365 ,D= 24.8029
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 87
Item count AFTER agenda pruned: 55
BestState:  COST= 2.88766 ,D= 24.8228
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
```

Item count BEFORE agenda pruned: 97
Item count AFTER agenda pruned: 44
BestState: COST= 3.7925 ,D= 23.8997
event(vp40,ewp,140923,140924)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 80
Item count AFTER agenda pruned: 47
BestState: COST= 3.92156 ,D= 24.0288
event(vp40,ewp,140927,140928)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 82
Item count AFTER agenda pruned: 49
BestState: COST= 3.9566 ,D= 24.0638
event(vp40,ewp,140928,140929)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 83
Item count AFTER agenda pruned: 51
BestState: COST= 3.99283 ,D= 24.1001
event(vp40,ewp,140929,140930)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 84
Item count AFTER agenda pruned: 53
BestState: COST= 4.03029 ,D= 24.1375
event(vp40,ewp,140930,140931)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 86
Item count AFTER agenda pruned: 57
BestState: COST= 6.32332 ,D= 24.6026
event(vp19,en,141242,141253)

event(vp40,ewp,140923,140924)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 92
Item count AFTER agenda pruned: 59
BestState: COST= 6.45237 ,D= 24.7316
event(vp19,en,141242,141253)
event(vp40,ewp,140927,140928)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 93
Item count AFTER agenda pruned: 60
BestState: COST= 6.48742 ,D= 24.7667
event(vp19,en,141242,141253)
event(vp40,ewp,140928,140929)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 93
Item count AFTER agenda pruned: 61
BestState: COST= 6.52365 ,D= 24.8029
event(vp19,en,141242,141253)
event(vp40,ewp,140929,140930)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 93
Item count AFTER agenda pruned: 62
BestState: COST= 6.56111 ,D= 24.8404
event(vp19,en,141242,141253)
event(vp40,ewp,140930,140931)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 94
Item count AFTER agenda pruned: 63
BestState: COST= 2.92389 ,D= 24.859

```
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 109
Item count AFTER agenda pruned: 53
BestState:  COST= 3.92389 ,D= 24.0311
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 88
Item count AFTER agenda pruned: 55
BestState:  COST= 3.95779 ,D= 24.065
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 89
Item count AFTER agenda pruned: 57
BestState:  COST= 3.99283 ,D= 24.1001
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 90
Item count AFTER agenda pruned: 59
BestState:  COST= 4.02906 ,D= 24.1363
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 92
Item count AFTER agenda pruned: 63
BestState:  COST= 6.45471 ,D= 24.734
  event(vp19,en,141242,141253)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
```

```
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 97
Item count AFTER agenda pruned: 64
BestState:  COST= 6.48861 ,D= 24.7679
  event(vp19,en,141242,141253)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 97
Item count AFTER agenda pruned: 65
BestState:  COST= 6.52365 ,D= 24.8029
  event(vp19,en,141242,141253)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 97
Item count AFTER agenda pruned: 66
BestState:  COST= 6.55988 ,D= 24.8392
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 98
Item count AFTER agenda pruned: 67
BestState:  COST= 2.92389 ,D= 24.859
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 109
Item count AFTER agenda pruned: 56
BestState:  COST= 3.82873 ,D= 23.9359
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 92
Item count AFTER agenda pruned: 59
BestState:  COST= 3.95779 ,D= 24.065
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 94
Item count AFTER agenda pruned: 61
BestState:  COST= 3.99283 ,D= 24.1001
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 95
Item count AFTER agenda pruned: 63
BestState:  COST= 4.02906 ,D= 24.1363
  event(vp40,ewp,140929,140930)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 96
Item count AFTER agenda pruned: 65
BestState:  COST= 4.06652 ,D= 24.1737
  event(vp40,ewp,140930,140931)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 98
Item count AFTER agenda pruned: 69
BestState:  COST= 6.35955 ,D= 24.6388
  event(vp19,en,141242,141253)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 104
Item count AFTER agenda pruned: 71
```

```
BestState:  COST= 6.48861 ,D= 24.7679
  event(vp19,en,141242,141253)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 105
Item count AFTER agenda pruned: 72
BestState:  COST= 6.52365 ,D= 24.8029
  event(vp19,en,141242,141253)
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 105
Item count AFTER agenda pruned: 73
BestState:  COST= 6.55988 ,D= 24.8392
  event(vp19,en,141242,141253)
  event(vp40,ewp,140929,140930)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 105
Item count AFTER agenda pruned: 74
BestState:  COST= 6.59734 ,D= 24.8766
  event(vp19,en,141242,141253)
  event(vp40,ewp,140930,140931)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 106
Item count AFTER agenda pruned: 75
BestState:  COST= 2.96136 ,D= 24.8965
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 121
Item count AFTER agenda pruned: 65
BestState:  COST= 3.96136 ,D= 24.0686
```

event(vp9,ewp,141018,141019)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 100
Item count AFTER agenda pruned: 67
BestState: COST= 3.99525 ,D= 24.1025
event(vp9,ewp,141019,141020)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 101
Item count AFTER agenda pruned: 69
BestState: COST= 4.03029 ,D= 24.1375
event(vp9,ewp,141020,141021)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 102
Item count AFTER agenda pruned: 71
BestState: COST= 4.06652 ,D= 24.1737
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 104
Item count AFTER agenda pruned: 75
BestState: COST= 6.49217 ,D= 24.7715
event(vp19,en,141242,141253)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 109
Item count AFTER agenda pruned: 76
BestState: COST= 6.52607 ,D= 24.8053
event(vp19,en,141242,141253)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140930,140931)

```
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 109
Item count AFTER agenda pruned: 77
BestState:  COST= 6.56111 ,D= 24.8404
  event(vp19,en,141242,141253)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 109
Item count AFTER agenda pruned: 78
BestState:  COST= 6.59734 ,D= 24.8766
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 110
Item count AFTER agenda pruned: 79
BestState:  COST= 6.82087 ,D= 25.1001
  event(vp48,en,141179,141190)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141018,141019)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 114
Item count AFTER agenda pruned: 79
BestState:  COST= 6.85476 ,D= 25.134
  event(vp48,en,141179,141190)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 113
Item count AFTER agenda pruned: 79
BestState:  COST= 6.8898 ,D= 25.1691
  event(vp48,en,141179,141190)
  event(vp40,ewp,140923,140924)
```

event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 112
Item count AFTER agenda pruned: 79
BestState: COST= 6.91754 ,D= 25.1968
event(vp47,ewp,141231,141232)
event(vp40,ewp,140923,140924)
event(vp9,ewp,141018,141019)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 119
Item count AFTER agenda pruned: 79
BestState: COST= 6.92604 ,D= 25.2053
event(vp48,en,141179,141190)
event(vp40,ewp,140923,140924)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 112
Item count AFTER agenda pruned: 79
BestState: COST= 6.94992 ,D= 25.2292
event(vp48,en,141179,141190)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 113
Item count AFTER agenda pruned: 79
BestState: COST= 6.95143 ,D= 25.2307
event(vp47,ewp,141231,141232)
event(vp40,ewp,140923,140924)
event(vp9,ewp,141019,141020)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 118
Item count AFTER agenda pruned: 79
BestState: COST= 6.98382 ,D= 25.2631
event(vp48,en,141179,141190)

event(vp40,ewp,140927,140928)
event(vp9,ewp,141019,141020)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 112
Item count AFTER agenda pruned: 79
BestState: COST= 6.98497 ,D= 25.2642
event(vp48,en,141179,141190)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140928,140929)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 112
Item count AFTER agenda pruned: 79
BestState: COST= 6.98648 ,D= 25.2657
event(vp47,ewp,141231,141232)
event(vp40,ewp,140923,140924)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 117
Item count AFTER agenda pruned: 79
BestState: COST= 7.01886 ,D= 25.2981
event(vp48,en,141179,141190)
event(vp40,ewp,140927,140928)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 111
Item count AFTER agenda pruned: 79
BestState: COST= 7.01886 ,D= 25.2981
event(vp48,en,141179,141190)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140928,140929)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 111
Item count AFTER agenda pruned: 79
BestState: COST= 7.0212 ,D= 25.3005

```
  event(vp48,en,141179,141190)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 111
Item count AFTER agenda pruned: 79
BestState:  COST= 7.02271 ,D= 25.302
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 117
Item count AFTER agenda pruned: 79
BestState:  COST= 7.0466 ,D= 25.3259
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140927,140928)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 118
Item count AFTER agenda pruned: 79
BestState:  COST= 7.05391 ,D= 25.3332
  event(vp48,en,141179,141190)
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 110
Item count AFTER agenda pruned: 79
BestState:  COST= 7.05509 ,D= 25.3344
  event(vp48,en,141179,141190)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 111
Item count AFTER agenda pruned: 79
```

```
BestState:  COST= 7.05509 ,D= 25.3344
  event(vp48,en,141179,141190)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 110
Item count AFTER agenda pruned: 79
BestState:  COST= 7.05866 ,D= 25.3379
  event(vp48,en,141179,141190)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 111
Item count AFTER agenda pruned: 79
BestState:  COST= 7.08049 ,D= 25.3598
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 117
Item count AFTER agenda pruned: 79
BestState:  COST= 7.08164 ,D= 25.3609
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 117
Item count AFTER agenda pruned: 79
BestState:  COST= 7.09014 ,D= 25.3694
  event(vp48,en,141179,141190)
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 110
```

```
Item count AFTER agenda pruned: 79
BestState:  COST= 7.09014 ,D= 25.3694
  event(vp48,en,141179,141190)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 109
Item count AFTER agenda pruned: 79
BestState:  COST= 7.09255 ,D= 25.3718
  event(vp48,en,141179,141190)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 110
Item count AFTER agenda pruned: 80
BestState:  COST= 7.11553 ,D= 25.3948
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 117
Item count AFTER agenda pruned: 80
BestState:  COST= 7.11553 ,D= 25.3948
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 117
Item count AFTER agenda pruned: 81
BestState:  COST= 7.11787 ,D= 25.3971
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
```

```
Item count BEFORE agenda pruned: 118
Item count AFTER agenda pruned: 81
BestState:  COST= 7.12637 ,D= 25.4056
  event(vp48,en,141179,141190)
  event(vp40,ewp,140929,140930)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 111
Item count AFTER agenda pruned: 81
BestState:  COST= 7.1276 ,D= 25.4069
  event(vp48,en,141179,141190)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 111
Item count AFTER agenda pruned: 82
BestState:  COST= 7.15058 ,D= 25.4299
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 118
Item count AFTER agenda pruned: 83
BestState:  COST= 7.15176 ,D= 25.431
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 120
Item count AFTER agenda pruned: 84
BestState:  COST= 7.15176 ,D= 25.431
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

```
---------------------------------------------------------
Item count BEFORE agenda pruned: 120
Item count AFTER agenda pruned: 85
BestState:  COST= 7.15533 ,D= 25.4346
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 122
Item count AFTER agenda pruned: 85
BestState:  COST= 7.16383 ,D= 25.4431
  event(vp48,en,141179,141190)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 115
Item count AFTER agenda pruned: 86
BestState:  COST= 7.18681 ,D= 25.4661
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 122
Item count AFTER agenda pruned: 87
BestState:  COST= 7.18922 ,D= 25.4685
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 123
Item count AFTER agenda pruned: 88
BestState:  COST= 7.21733 ,D= 25.4966
  event(vp47,ewp,141230,141231)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 126
Item count AFTER agenda pruned: 89
BestState:  COST= 7.22304 ,D= 25.5023
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140929,140930)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 124
Item count AFTER agenda pruned: 90
BestState:  COST= 7.22427 ,D= 25.5035
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 125
Item count AFTER agenda pruned: 90
BestState:  COST= 7.2605 ,D= 25.5398
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 125
Item count AFTER agenda pruned: 91
BestState:  COST= 7.45512 ,D= 25.7344
  event(vp47,ewp,141230,141231)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 126
Item count AFTER agenda pruned: 92
BestState:  COST= 9.35168 ,D= 25.803
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141018,141019)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
```

```
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 126
Item count AFTER agenda pruned: 92
BestState:  COST= 9.38557 ,D= 25.8369
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 125
Item count AFTER agenda pruned: 92
BestState:  COST= 9.42062 ,D= 25.872
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 124
Item count AFTER agenda pruned: 92
BestState:  COST= 9.44836 ,D= 25.8997
  event(vp19,en,141242,141253)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141018,141019)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 131
Item count AFTER agenda pruned: 92
BestState:  COST= 9.45686 ,D= 25.9082
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------
Item count BEFORE agenda pruned: 124
Item count AFTER agenda pruned: 92
BestState:  COST= 9.48074 ,D= 25.9321
```

event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 125
Item count AFTER agenda pruned: 92
BestState: COST= 9.48225 ,D= 25.9336
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp40,ewp,140923,140924)
event(vp9,ewp,141019,141020)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 130
Item count AFTER agenda pruned: 92
BestState: COST= 7.66938 ,D= 25.9487
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 128
Item count AFTER agenda pruned: 93
BestState: COST= 9.51463 ,D= 25.966
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp40,ewp,140927,140928)
event(vp9,ewp,141019,141020)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 125
Item count AFTER agenda pruned: 93
BestState: COST= 9.51579 ,D= 25.9671
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140928,140929)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 125
Item count AFTER agenda pruned: 93
BestState:  COST= 9.5173 ,D= 25.9686
  event(vp19,en,141242,141253)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 130
Item count AFTER agenda pruned: 93
BestState:  COST= 9.54967 ,D= 26.001
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 124
Item count AFTER agenda pruned: 93
BestState:  COST= 9.54967 ,D= 26.001
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 124
Item count AFTER agenda pruned: 93
BestState:  COST= 9.55202 ,D= 26.0034
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 124
Item count AFTER agenda pruned: 93
BestState:  COST= 9.55353 ,D= 26.0049
  event(vp19,en,141242,141253)
  event(vp47,ewp,141231,141232)
```

event(vp40,ewp,140923,140924)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 130
Item count AFTER agenda pruned: 93
BestState: COST= 9.57742 ,D= 26.0288
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140927,140928)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 131
Item count AFTER agenda pruned: 93
BestState: COST= 9.58472 ,D= 26.0361
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp40,ewp,140928,140929)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 123
Item count AFTER agenda pruned: 93
BestState: COST= 9.58591 ,D= 26.0373
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140929,140930)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 123
Item count AFTER agenda pruned: 93
BestState: COST= 9.58591 ,D= 26.0373
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp40,ewp,140927,140928)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 124
Item count AFTER agenda pruned: 92
BestState:  COST= 9.58948 ,D= 26.0408
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 123
Item count AFTER agenda pruned: 92
BestState:  COST= 9.61131 ,D= 26.0627
  event(vp19,en,141242,141253)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 129
Item count AFTER agenda pruned: 92
BestState:  COST= 9.61246 ,D= 26.0638
  event(vp19,en,141242,141253)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 129
Item count AFTER agenda pruned: 92
BestState:  COST= 9.62096 ,D= 26.0723
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 121
Item count AFTER agenda pruned: 92
BestState:  COST= 9.62096 ,D= 26.0723
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
```

event(vp40,ewp,140928,140929)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 122
Item count AFTER agenda pruned: 91
BestState: COST= 9.62337 ,D= 26.0747
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 121
Item count AFTER agenda pruned: 91
BestState: COST= 9.64635 ,D= 26.0977
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140928,140929)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 127
Item count AFTER agenda pruned: 91
BestState: COST= 9.64635 ,D= 26.0977
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp40,ewp,140927,140928)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 127
Item count AFTER agenda pruned: 91
BestState: COST= 9.64869 ,D= 26.1
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140929,140930)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 127
Item count AFTER agenda pruned: 91
BestState: COST= 9.65719 ,D= 26.1085
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp40,ewp,140929,140930)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 120
Item count AFTER agenda pruned: 90
BestState: COST= 9.65842 ,D= 26.1098
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp9,ewp,141020,141021)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 119
Item count AFTER agenda pruned: 90
BestState: COST= 9.6814 ,D= 26.1328
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp40,ewp,140928,140929)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 125
Item count AFTER agenda pruned: 90
BestState: COST= 9.68258 ,D= 26.1339
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140929,140930)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 125
Item count AFTER agenda pruned: 90
BestState: COST= 9.68258 ,D= 26.1339
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)

event(vp40,ewp,140927,140928)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 126
Item count AFTER agenda pruned: 89
BestState: COST= 9.68615 ,D= 26.1375
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 125
Item count AFTER agenda pruned: 88
BestState: COST= 9.69465 ,D= 26.146
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 117
Item count AFTER agenda pruned: 88
BestState: COST= 9.71763 ,D= 26.169
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp40,ewp,140928,140929)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 123
Item count AFTER agenda pruned: 87
BestState: COST= 9.72004 ,D= 26.1714
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

```
---------------------------------------------------------
Item count BEFORE agenda pruned: 122
Item count AFTER agenda pruned: 87
BestState:  COST= 9.74815 ,D= 26.1995
  event(vp19,en,141242,141253)
  event(vp47,ewp,141230,141231)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 124
Item count AFTER agenda pruned: 87
BestState:  COST= 9.75386 ,D= 26.2052
  event(vp19,en,141242,141253)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140929,140930)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 121
Item count AFTER agenda pruned: 86
BestState:  COST= 9.75509 ,D= 26.2064
  event(vp19,en,141242,141253)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 120
Item count AFTER agenda pruned: 85
BestState:  COST= 9.79131 ,D= 26.2427
  event(vp47,ewp,141231,141232)
  event(vp19,en,141242,141253)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 119
Item count AFTER agenda pruned: 84
BestState:  COST= 9.98594 ,D= 26.4373
  event(vp19,en,141242,141253)
  event(vp47,ewp,141230,141231)
```

```
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 118
Item count AFTER agenda pruned: 83
BestState:  COST= 10.0148 ,D= 26.4662
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141018,141019)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 120
Item count AFTER agenda pruned: 82
BestState:  COST= 10.0487 ,D= 26.5001
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 118
Item count AFTER agenda pruned: 81
BestState:  COST= 10.0838 ,D= 26.5351
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 116
Item count AFTER agenda pruned: 80
BestState:  COST= 10.12 ,D= 26.5714
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 115
Item count AFTER agenda pruned: 79
BestState:  COST= 10.1439 ,D= 26.5952
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140927,140928)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 115
Item count AFTER agenda pruned: 78
BestState:  COST= 10.1778 ,D= 26.6292
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 113
Item count AFTER agenda pruned: 77
BestState:  COST= 10.1789 ,D= 26.6303
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 112
Item count AFTER agenda pruned: 77
BestState:  COST= 10.2002 ,D= 26.6516
  event(vp19,en,141242,141253)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 112
Item count AFTER agenda pruned: 76
BestState:  COST= 10.2128 ,D= 26.6642
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
```

```
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140928,140929)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 110
Item count AFTER agenda pruned: 75
BestState:  COST= 10.2128 ,D= 26.6642
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 109
Item count AFTER agenda pruned: 74
BestState:  COST= 10.2152 ,D= 26.6665
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 108
Item count AFTER agenda pruned: 73
BestState:  COST= 10.2479 ,D= 26.6992
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 106
Item count AFTER agenda pruned: 72
BestState:  COST= 10.2491 ,D= 26.7004
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 105
Item count AFTER agenda pruned: 71
BestState:  COST= 10.2491 ,D= 26.7004
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 105
Item count AFTER agenda pruned: 70
BestState:  COST= 10.2526 ,D= 26.704
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 104
Item count AFTER agenda pruned: 69
BestState:  COST= 10.2841 ,D= 26.7355
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 102
Item count AFTER agenda pruned: 69
BestState:  COST= 10.2841 ,D= 26.7355
  event(vp47,ewp,141231,141232)
  event(vp48,en,141179,141190)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140929,140930)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 101
Item count AFTER agenda pruned: 68
BestState:  COST= 10.2865 ,D= 26.7379
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
```

event(vp9,ewp,141019,141020)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 101
Item count AFTER agenda pruned: 68
BestState: COST= 10.3146 ,D= 26.766
event(vp48,en,141179,141190)
event(vp47,ewp,141230,141231)
event(vp40,ewp,140923,140924)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 103
Item count AFTER agenda pruned: 67
BestState: COST= 10.3203 ,D= 26.7717
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp40,ewp,140929,140930)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 99
Item count AFTER agenda pruned: 66
BestState: COST= 10.3216 ,D= 26.7729
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141020,141021)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 98
Item count AFTER agenda pruned: 65
BestState: COST= 10.3578 ,D= 26.8092
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

```
---------------------------------------------------------------
Item count BEFORE agenda pruned: 97
Item count AFTER agenda pruned: 65
BestState:  COST= 10.5524 ,D= 27.0038
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 97
Item count AFTER agenda pruned: 63
BestState:  COST= 12.5457 ,D= 27.1691
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141018,141019)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 99
Item count AFTER agenda pruned: 60
BestState:  COST= 12.5796 ,D= 27.203
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 95
Item count AFTER agenda pruned: 59
BestState:  COST= 10.7667 ,D= 27.218
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 92
Item count AFTER agenda pruned: 57
BestState:  COST= 12.6146 ,D= 27.238
```

```
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141020,141021)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 91
Item count AFTER agenda pruned: 54
BestState:  COST= 12.6508 ,D= 27.2743
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 88
Item count AFTER agenda pruned: 51
BestState:  COST= 12.6747 ,D= 27.2981
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141018,141019)
  event(vp40,ewp,140927,140928)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 86
Item count AFTER agenda pruned: 48
BestState:  COST= 12.7086 ,D= 27.332
  event(vp47,ewp,141231,141232)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp40,ewp,140927,140928)
  event(vp9,ewp,141019,141020)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 82
Item count AFTER agenda pruned: 45
BestState:  COST= 12.7098 ,D= 27.3332
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
```

event(vp47,ewp,141231,141232)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140928,140929)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 79
Item count AFTER agenda pruned: 42
BestState: COST= 12.7437 ,D= 27.3671
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp40,ewp,140927,140928)
event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 75
Item count AFTER agenda pruned: 39
BestState: COST= 12.7437 ,D= 27.3671
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140928,140929)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 72
Item count AFTER agenda pruned: 36
BestState: COST= 12.746 ,D= 27.3694
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140929,140930)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 69
Item count AFTER agenda pruned: 33
BestState: COST= 12.7787 ,D= 27.4021
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp40,ewp,140928,140929)

event(vp9,ewp,141020,141021)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 65
Item count AFTER agenda pruned: 30
BestState: COST= 12.7799 ,D= 27.4033
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp40,ewp,140927,140928)
event(vp9,ewp,141021,141022)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 63
Item count AFTER agenda pruned: 27
BestState: COST= 12.7799 ,D= 27.4033
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141019,141020)
event(vp40,ewp,140929,140930)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 59
Item count AFTER agenda pruned: 24
BestState: COST= 12.7834 ,D= 27.4069
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141231,141232)
event(vp9,ewp,141018,141019)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 57
Item count AFTER agenda pruned: 22
BestState: COST= 12.8149 ,D= 27.4384
event(vp19,en,141242,141253)
event(vp47,ewp,141231,141232)
event(vp48,en,141179,141190)
event(vp9,ewp,141020,141021)
event(vp40,ewp,140929,140930)
event(vp50,ewp,141046,141047)

```
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 53
Item count AFTER agenda pruned: 19
BestState:  COST= 12.8149 ,D= 27.4384
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140928,140929)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 51
Item count AFTER agenda pruned: 16
BestState:  COST= 12.8173 ,D= 27.4408
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141019,141020)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 48
Item count AFTER agenda pruned: 14
BestState:  COST= 12.8455 ,D= 27.4689
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp40,ewp,140923,140924)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------------
Item count BEFORE agenda pruned: 48
Item count AFTER agenda pruned: 11
BestState:  COST= 12.8512 ,D= 27.4746
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp40,ewp,140929,140930)
  event(vp9,ewp,141021,141022)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)
```

```
---------------------------------------------------------
Item count BEFORE agenda pruned: 42
Item count AFTER agenda pruned: 8
BestState:  COST= 12.8524 ,D= 27.4758
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141020,141021)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 39
Item count AFTER agenda pruned: 5
BestState:  COST= 12.8886 ,D= 27.5121
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141231,141232)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 36
Item count AFTER agenda pruned: 3
BestState:  COST= 13.0832 ,D= 27.7067
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141230,141231)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 34
Item count AFTER agenda pruned: 1
BestState:  COST= 13.2975 ,D= 27.9209
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

---------------------------------------------------------
Item count BEFORE agenda pruned: 33
```

Item count AFTER agenda pruned: 2
BestState: COST= 18.6509 ,D= 31.4464
event(vp40,ewn,140949,140950)
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 32
Item count AFTER agenda pruned: 1
BestState: COST= 18.6817 ,D= 31.4772
event(vp40,ewn,140948,140949)
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 31
Item count AFTER agenda pruned: 6
BestState: COST= 28.9573 ,D= 39.9248
event(vp9,en,141095,141106)
event(vp40,ewn,140948,140949)
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 32
Item count AFTER agenda pruned: 5
BestState: COST= 29.2432 ,D= 40.2108
event(vp9,en,141102,141113)
event(vp40,ewn,140948,140949)
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)

```
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 31
Item count AFTER agenda pruned: 4
BestState:  COST= 29.2432 ,D= 40.2108
  event(vp9,en,141088,141099)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 30
Item count AFTER agenda pruned: 3
BestState:  COST= 29.5317 ,D= 40.4992
  event(vp50,en,141137,141148)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 30
Item count AFTER agenda pruned: 2
BestState:  COST= 30.1185 ,D= 41.0861
  event(vp9,en,141081,141092)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 28
Item count AFTER agenda pruned: 1
BestState:  COST= 30.2947 ,D= 41.2623
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
```

event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 27
Item count AFTER agenda pruned: 6
BestState: COST= 34.1768 ,D= 43.3164
event(vp50,ewn,141070,141071)
event(vp9,ewn,141039,141040)
event(vp40,ewn,140948,140949)
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 27
Item count AFTER agenda pruned: 5
BestState: COST= 34.2522 ,D= 43.3918
event(vp50,ewn,141069,141070)
event(vp9,ewn,141039,141040)
event(vp40,ewn,140948,140949)
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 26
Item count AFTER agenda pruned: 4
BestState: COST= 34.3427 ,D= 43.4824
event(vp50,ewn,141068,141069)
event(vp9,ewn,141039,141040)
event(vp40,ewn,140948,140949)
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)

event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 25
Item count AFTER agenda pruned: 3
BestState: COST= 34.4495 ,D= 43.5891
event(vp50,ewn,141067,141068)
event(vp9,ewn,141039,141040)
event(vp40,ewn,140948,140949)
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 24
Item count AFTER agenda pruned: 2
BestState: COST= 35.0625 ,D= 44.2021
event(vp50,ewn,141063,141064)
event(vp9,ewn,141039,141040)
event(vp40,ewn,140948,140949)
event(vp19,en,141242,141253)
event(vp48,en,141179,141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 22
Item count AFTER agenda pruned: 1
BestState: COST= 35.2694 ,D= 44.409
event(vp50,ewn,141062,141063)
event(vp9,ewn,141039,141040)
event(vp40,ewn,140948,140949)
event(vp19,en,141242,141253)
event(vp48,en,141179 141190)
event(vp47,ewp,141229,141230)
event(vp9,ewp,141021,141022)
event(vp40,ewp,140930,140931)
event(vp50,ewp,141046,141047)
event(vp47,en,141207,141218)
event(vp46,ewn,140893,140894)

Item count BEFORE agenda pruned: 21
Item count AFTER agenda pruned: 3
BestState: COST= 42.4002 ,D= 49.7119

```
  event(vp50,en,141137,141148)
  event(vp50,ewn,141062,141063)
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 20
Item count AFTER agenda pruned: 2
BestState:  COST= 42.9459 ,D= 50.2576
  event(vp9,en,141102,141113)
  event(vp50,ewn,141062,141063)
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 18
Item count AFTER agenda pruned: 1
BestState:  COST= 43.2931 ,D= 50.6048
  event(vp9,en,141095,141106)
  event(vp50,ewn,141062,141063)
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-------------------------------------------------------------
Item count BEFORE agenda pruned: 17
Item count AFTER agenda pruned: 1
BestState:  COST= 47.107 ,D= 52.5908
  event(vp50,en,141137,141148)
  event(vp9,en,141095,141106)
  event(vp50,ewn,141062,141063)
```

```
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 14
Item count AFTER agenda pruned: 2
BestState:  COST= 59.4271 ,D= 63.0829
  event(vp40,en,140990,141001)
  event(vp50,en,141137,141148)
  event(vp9,en,141095,141106)
  event(vp50,ewn,141062,141063)
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 10
Item count AFTER agenda pruned: 1
BestState:  COST= 60.4121 ,D= 64.068
  event(vp40,en,141046,141057)
  event(vp50,en,141137,141148)
  event(vp9,en,141095,141106)
  event(vp50,ewn,141062,141063)
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

------------------------------------------------------------
Item count BEFORE agenda pruned: 9
Item count AFTER agenda pruned: 2
BestState:  COST= 69.9102 ,D= 71.738
  event(vp46,en,140962,140973)
```

```
  event(vp40,en,141046,141057)
  event(vp50,en,141137,141148)
  event(vp9,en,141095,141106)
  event(vp50,ewn,141062,141063)
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------------
Item count BEFORE agenda pruned: 7
Item count AFTER agenda pruned: 1
BestState:  COST= 70.3666 ,D= 72.1945
  event(vp46,en,140955,140966)
  event(vp40,en,141046,141057)
  event(vp50,en,141137,141148)
  event(vp9,en,141095,141106)
  event(vp50,ewn,141062,141063)
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
  event(vp46,ewn,140893,140894)

-----------------------------------------------------------------
Item count BEFORE agenda pruned: 6
Item count AFTER agenda pruned: 1
BestState:  COST= 266.993 ,D= 266.993
  event(vp47,ewn,141245,141246)
  event(vp46,en,140955,140966)
  event(vp40,en,141046,141057)
  event(vp50,en,141137,141148)
  event(vp9,en,141095,141106)
  event(vp50,ewn,141062,141063)
  event(vp9,ewn,141039,141040)
  event(vp40,ewn,140948,140949)
  event(vp19,en,141242,141253)
  event(vp48,en,141179,141190)
  event(vp47,ewp,141229,141230)
  event(vp9,ewp,141021,141022)
  event(vp40,ewp,140930,140931)
  event(vp50,ewp,141046,141047)
  event(vp47,en,141207,141218)
```

```
  event(vp46,ewn,140893,140894)

************************************************************
0 incompletely examined state(s) and 245 examined state(s)
SEARCH 2 RESULTS:
The process for search2 completed in: 15881.5 cpu seconds.
************************************************************
==> Successors generated during search2: 8498
==> Average process time for successors in search2: 1.86885 cpu seconds.
************************************************************

Almost there, scheduling the LAST third of the schedule....

------------------------------------------------------------
Item count BEFORE agenda pruned: 33
Item count AFTER agenda pruned: 22
BestState:  COST= 0.449329 ,D= 26.0403
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 51
Item count AFTER agenda pruned: 29
BestState:  COST= 0.898658 ,D= 24.6617
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 60
Item count AFTER agenda pruned: 13
BestState:  COST= 1.44747 ,D= 23.3826
  event(vp48,ec,140899,140903)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 42
Item count AFTER agenda pruned: 15
BestState:  COST= 1.77383 ,D= 23.709
  event(vp48,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

------------------------------------------------------------
Item count BEFORE agenda pruned: 43
Item count AFTER agenda pruned: 16
BestState:  COST= 1.77383 ,D= 23.709
  event(vp19,ec,140913,140917)
  event(vp40,ewc,141075,141078)
  event(vp47,ewm,140893,140896)

.
.
.
202 BestStates skipped...
```

```
------------------------------------------------------------
Item count BEFORE agenda pruned: 3
Item count AFTER agenda pruned: 1
BestState:  COST= 404.803 ,D= 404.803
  event(vp9,ec,141235,141239)
  event(vp40,ec,141137,141141)
  event(vp46,ec,141109,141113)
  event(vp50,ewm,141256,141259)
  event(vp9,ewm,141201,141204)
  event(vp40,ewm,141117,141120)
  event(vp46,ewm,141074,141077)
  event(vp46,ewc,141047,141050)
  event(vp40,ewc,141081,141084)
  event(vp19,ec,140920,140924)
  event(vp48,ec,140899,140903)
  event(vp50,ewc,141222,141225)
  event(vp47,ec,140927,140931)
  event(vp9,ewc,141173,141176)
  event(vp47,ewm,140893,140896)

************************************************************
0 incompletely examined state(s) and 207 examined state(s)
SEARCH 3 RESULTS:
The process for search3 completed in: 6809.23 cpu seconds.
************************************************************
==> Successors generated during search3: 5735
==> Average process time for successors in search3: 1.18731 cpu seconds.
************************************************************

The process for flagging_dropdeaddates completed in: 0.766602 cpu seconds.
************************************************************
```

TRAINING SCHEDULE FOR FISCAL YEAR 1986

event(vp46,ewn,[1,oct,1985],[2,oct,1985])
event(vp47,ewm,[1,oct,1985],[4,oct,1985])
event(vp48,ec,[7,oct,1985],[11,oct,1985])
event(vp46,tr1a,[16,oct,1985],[30,nov,1985])
event(vp19,ec,[28,oct,1985],[1,nov,1985])
event(vp47,ec,[4,nov,1985],[8,nov,1985])
event(vp40,ewp,[7,nov,1985],[8,nov,1985])
event(vp40,ewn,[25,nov,1985],[26,nov,1985])
event(vp40,tr1a,[1,dec,1985],[31,dec,1985])
event(vp46,en,[2,dec,1985],[13,dec,1985])
event(vp46,tr1b,[1,jan,1986],[31,jan,1986])
event(vp40,tr1b,[1,feb,1986],[28,feb,1986])
event(vp9,ewp,[6,feb,1986],[7,feb,1986])
event(vp9,ewn,[24,feb,1986],[25,feb,1986])
event(vp9,tr1a,[1,mar,1986],[31,mar,1986])
event(vp50,ewp,[3,mar,1986],[4,mar,1986])
event(vp40,en,[3,mar,1986],[14,mar,1986])
event(vp46,ewc,[4,mar,1986],[7,mar,1986])
event(vp50,ewn,[19,mar,1986],[20,mar,1986])
event(vp46,ewm,[31,mar,1986],[3,apr,1986])
event(vp50,tr1a,[1,apr,1986],[15,may,1986])
event(vp40,ewc,[7,apr,1986],[10,apr,1986])
event(vp9,en,[21,apr,1986],[2,may,1986])
event(vp46,ec,[5,may,1986],[9,may,1986])
event(vp40,ewm,[13,may,1986],[16,may,1986])
event(vp9,tr1b,[16,may,1986],[30,jun,1986])
event(vp50,en,[2,jun,1986],[13,jun,1986])
event(vp40,ec,[2,jun,1986],[6,jun,1986])
event(vp50,tr1b,[1,jul,1986],[15,aug,1986])
event(vp9,ewc,[8,jul,1986],[11,jul,1986])
event(vp48,en,[14,jul,1986],[25,jul,1986])
event(vp9,ewm,[5,aug,1986],[8,aug,1986])
event(vp47,en,[11,aug,1986],[22,aug,1986])
event(vp48,tr1a,[16,aug,1986],[30,sep,1986])
event(vp50,ewc,[26,aug,1986],[29,aug,1986])
event(vp47,ewp,[2,sep,1986],[3,sep,1986])
event(vp9,ec,[8,sep,1986],[12,sep,1986])
event(vp19,en,[15,sep,1986],[26,sep,1986])
event(vp47,ewn,[18,sep,1986],[19,sep,1986])
event(vp50,ewm,[29,sep,1986],[2,oct,1986])

LIST OF REFERENCES

1. O'Brien, J. J., *Scheduling Handbook,* pp. 2-5, McGraw-Hill Book Company, New York, 1969.
2. Baker, K. R., *Introduction to Sequencing and Scheduling,* pp. 2-6, John Wiley & Sons, Incorporated, New York, 1974.
3. Urwick, S., "Development of Industrial Engineering," in *Industrial Engineering*, ed. Maynard, McGraw-Hill Book Company, New York, 1963.
4. Martino, R. L., *Resources Management,* pp. 69-72, Management Development Institute, Incorporated, Wayne, Pennsylvania, 1968.
5. Davis, R., "Amplifying Expertise with Expert Systems," in *The AI Business*, ed. Winston, Patrick H., p. 19, The MIT Press, Cambridge, Massachusetts, 1984.
6. Nilsson, N. J., *Principles of Artificial Intelligence,* pp. 6-419, Tioga Publishing Company, Palo Alto, California, 1980.
7. Rowe, N. C., *Artificial Intelligence Through Prolog,* Prentice-Hall, Inglewood Cliffs, New Jersey, 1988.
8. Telephone conversation between George S. Sanford, Jr., Lieutenant Commander, USN, Weapons Officer, Commander, Patrol Wings Pacific, NAS Moffett Field, California, and the author, 13 May 1988.
9. Interview between George S. Sanford, Jr., Lieutaenant Commander, USN, Weapons Officer, Commander Patrol Wings Pacific, NAS Moffett Field, California, and the author, 11 April 1988.
10. Interview between Bill Eckert, Lieutaenant Commander, USN, Training Officer, Commander Patrol Wing Eleven, NAS Jacksonville, Florida, and the author, 16 August 1987.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Chief of Naval Operations Director, Information Systems (OP-945) Navy Department Washington, D.C. 20350-2000	1
4.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, California 93943-5000	2
5.	Curricular Officer, Code 37 Computer Technology Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor Neil C. Rowe, Code 52Rp Computer Science Department Naval Postgraduate School Monterey, California 93943-5000	2
7.	Professor Robert B. McGhee, Code 52Mz Computer Science Department Naval Postgraduate School Monterey, California 93943-5000	1
8.	Commanding Officer Patrol Squadron NINE Naval Air Station Moffett Field, California 94035	1

9.	LCDR George S. Sanford, Jr. Weapons Training Officer Patrol Wing5 Pacific Naval Air Station Moffett Field, California 94035	1
10.	Dr. Hank Smith Education Coordinator Patrol Squadron THIRTY-ONE Naval Air Station Moffett Field, California 94035	1
11.	Mrs. Jean C. Hooper Summer Chase Glynwood Apartments C-5 McQueen-Smith Road Prattville, Alabama 36067	1
12.	Mr.and Mrs. David V. Hutson 599-C Michelson Road Monterey, California 93940	1